

YX 2320
p.c-A

TOPS-20 User Utilities Guide

AA-D850A-TM, AD-D850A-T1

January 1982

This manual describes utility programs available to both privileged and nonprivileged users of the TOPS-20 operating system.

OPERATING SYSTEM:

TOPS-20 V4

SOFTWARE:

MAKLIB V2B

Software and manuals should be ordered by title and order number. In the United States, send orders to the nearest distribution center. Outside the United States, orders should be directed to the nearest DIGITAL Field Sales Office or representative.

Northeast/Mid-Atlantic Region

Digital Equipment Corporation
PO Box CS2008
Nashua, New Hampshire 03061
Telephone: (603)884-6660

Central Region

Digital Equipment Corporation
Accessories and Supplies Center
1050 East Remington Road
Schaumburg, Illinois 60195
Telephone: (312)640-5612

Western Region

Digital Equipment Corporation
Accessories and Supplies Center
632 Caribbean Drive
Sunnyvale, California 94086
Telephone: (408)734-4915

digital equipment corporation • marlboro, massachusetts

January 1980
Updated, January 1982

Copyright ©, 1980, 1982, Digital Equipment Corporation. All Rights Reserved.

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by DIGITAL or its affiliated companies.

The following are trademarks of Digital Equipment Corporation:

| | | |
|----------------|--------------|---------|
| DEC | DECnet | IAS |
| DECUS | DECsystem-10 | MASSBUS |
| DECSYSTEM-20 | PDT | PDP |
| DECwriter | RSTS | UNIBUS |
| DIBOL | RSX | VAX |
| EduSystem | VMS | VT |
| digital | | |

The postage-prepaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

CONTENTS

| | Page |
|--|------|
| PREFACE | vii |
| CHAPTER 1 INTRODUCTION TO TOPS-20 USER UTILITIES | 1-1 |
| 1.1 INVOKING THE UTILITIES | 1-1 |
| 1.2 TYPING FILE SPECIFICATIONS | 1-1 |
| CHAPTER 2 THE MAIL PROGRAM | 2-1 |
| 2.1 INTRODUCTION | 2-1 |
| 2.2 RUNNING MAIL | 2-1 |
| 2.3 OPTIONS TO THE MAIL PROCEDURE | 2-4 |
| 2.4 MAIL MESSAGES | 2-6 |
| 2.5 TECHNICAL NOTES | 2-9 |
| CHAPTER 3 THE RDMAIL PROGRAM | 3-1 |
| 3.1 INTRODUCTION | 3-1 |
| 3.2 RUNNING RDMAIL | 3-2 |
| 3.2.1 Reading Mail Using RDMAIL Switches | 3-4 |
| 3.3 RDMAIL MESSAGES | 3-7 |
| CHAPTER 4 THE FILCOM PROGRAM | 4-1 |
| 4.1 INTRODUCTION | 4-1 |
| 4.2 RUNNING FILCOM | 4-1 |
| 4.2.1 Comparing ASCII Files | 4-3 |
| 4.2.2 Comparing Binary Files | 4-7 |
| 4.3 FILCOM SWITCHES | 4-11 |
| 4.4 FILCOM MESSAGES | 4-12 |
| CHAPTER 5 THE CREF PROGRAM | 5-1 |
| 5.1 INTRODUCTION | 5-1 |
| 5.2 RUNNING CREF | 5-1 |
| 5.2.1 Creating .CRF Files with COMPILE | 5-1 |
| 5.2.2 Producing Cross-Reference Listings | 5-2 |
| 5.3 CREF EXAMPLES | 5-6 |
| 5.4 CREF MESSAGES | 5-10 |
| 5.5 TECHNICAL NOTES | 5-14 |
| CHAPTER 6 THE MAKLIB PROGRAM | 6-1 |
| 6.1 INTRODUCTION | 6-1 |
| 6.2 RUNNING MAKLIB | 6-3 |
| 6.2.1 Running MAKLIB to Obtain Information About Libraries | 6-4 |
| 6.2.2 Running MAKLIB to Manipulate Libraries | 6-7 |
| 6.2.3 Running MAKLIB to Modify Libraries | 6-19 |
| 6.2.4 Running MAKLIB to Edit Libraries | 6-20 |
| 6.3 MAKLIB SWITCH OPTIONS | 6-27 |

CONTENTS (CONT.)

| | Page |
|--|-------------|
| 6.4 MAKLIB MESSAGES | 6-28 |
| 6.5 TECHNICAL NOTES | 6-41 |
| 6.5.1 Format of TRACE Block Data (REL Block Type 1060) | 6-41 |
| 6.5.2 Format of Code Insertion | 6-43 |
| CHAPTER 7 THE DUMPER PROGRAM | 7-1 |
| 7.1 INTRODUCTION | 7-1 |
| 7.2 FEATURES | 7-2 |
| 7.3 USING TAPES WITH AND WITHOUT TAPE DRIVE ALLOCATION | 7-3 |
| 7.4 RUNNING DUMPER | 7-6 |
| 7.5 THE NONPRIVILEGED USER | 7-7 |
| 7.5.1 Setting the Status of Operation | 7-7 |
| 7.5.2 Positioning the Tape | 7-20 |
| 7.5.3 Interacting with Tape Files | 7-23 |
| 7.5.4 Marking Files to be Archived | 7-31 |
| 7.5.5 Additional Examples | 7-31 |
| 7.6 THE PRIVILEGED USER | 7-42 |
| 7.6.1 Backing Up System Files and/or Other Users' Files | 7-44 |
| 7.6.2 Restoring Files and Directories from System Backup Tapes | 7-46 |
| 7.6.3 Archiving Marked Files | 7-46 |
| 7.6.4 Migrating Files | 7-49 |
| 7.6.5 Retrieving or Restoring Archived and Migrated Files | 7-50 |
| 7.7 DUMPER COMMANDS | 7-51 |
| 7.8 DUMPER MESSAGES | 7-57 |
| CHAPTER 8 THE PLEASE PROGRAM | 8-1 |
| 8.1 INTRODUCTION | 8-1 |
| 8.2 RUNNING PLEASE | 8-1 |
| 8.2.1 Running PLEASE at TOPS-20 Command Level | 8-2 |
| 8.2.2 Running PLEASE at Program Command Level | 8-4 |
| 8.3 PLEASE MESSAGES | 8-6 |
| INDEX | Index-1 |

FIGURES

| | | |
|------------|--|------|
| FIGURE 2-1 | The Relationship Between MAIL and MAILER | 2-9 |
| 6-1 | Generation of an .EXE File | 6-1 |
| 6-2 | Generation of a Library | 6-2 |
| 6-3 | Function of /APPEND | 6-8 |
| 6-4 | Function of /DELETE | 6-10 |
| 6-5 | Function of /EXTRACT | 6-12 |
| 6-6 | One Function of /INSERT | 6-14 |
| 6-7 | One Function of /INSERT | 6-16 |
| 6-8 | Function of /REPLACE | 6-18 |
| 6-9 | Order of Pseudo-ops in a .FIX File | 6-24 |

CONTENTS (CONT.)

Page

TABLES





| | | | |
|-------|-----|--|------|
| TABLE | 3-1 | RDMAIL Switches | 3-4 |
| | 4-1 | Special File Types Recognized by FILCOM | 4-2 |
| | 4-2 | FILCOM Switches | 4-11 |
| | 4-3 | Reasons for File Access Errors | 4-14 |
| | 5-1 | CREF Switch Options | 5-4 |
| | 5-2 | Reasons for File Access Errors | 5-13 |
| | 5-3 | Error Status Codes | 5-14 |
| | 5-4 | Beginning and Ending Control Characters | 5-15 |
| | 5-5 | Symbol-Definition Control Characters | 5-16 |
| | 5-6 | Character-Count-Definition Characters | 5-17 |
| | 6-1 | MAKLIB Switches | 6-27 |
| | 7-1 | Status-Setting Commands | 7-8 |
| | 7-2 | Tape-Positioning Commands | 7-20 |
| | 7-3 | Action Commands | 7-24 |
| | 7-4 | File Descriptor Block (FDB) Entries Checked by DUMPER | 7-28 |

PREFACE

The TOPS-20 User Utilities Guide is intended for both the privileged and the nonprivileged user who needs information on utility programs that run on the TOPS-20 operating system. Before you use this manual, you should be familiar with the information contained in Getting Started with TOPS-20, the TOPS-20 User's Guide, and the TOPS-20 Commands Reference Manual.

This document provides detailed information on the following TOPS-20 utility programs: MAIL, RDMAIL, FILCOM, CREF, MAKLIB, DUMPER, and PLEASE. The manual contains tutorial and reference material in each chapter to accommodate both the novice and the experienced user.

The following conventions are used throughout the TOPS-20 User Utilities Guide:

| | |
|---|---|
|  | Indicates when you should press the RETURN key (on some terminals the key labeled CR) |
|  | Indicates when you should press the ESC key (on some terminals the key labeled ALT) |
|  | Indicates when you should press the DELETE key |
|  | Indicates when you should hold down the CTRL key and at the same time type the letter x |
| file spec | Indicates a file specification |
| red print | Indicates anything you type or are expected to type on your terminal. |

The following TOPS-20 documents are referenced in this manual:

Getting Started With TOPS-20 (AA-4187D-TM)

TOPS-20 User's Guide (AA-4179C-TM)

TOPS-20 Commands Reference Manual (AA-5115B-TM)

TOPS-20 Operator's Guide (AA-4176D-TM)

TOPS-20 Monitor Calls Reference Manual (AA-4166D-TM)

TOPS-20 LINK Reference Manual (AA-4183A-TM)

TOPS-20 MACRO ASSEMBLER Reference Manual (AA-4159C-TM)

TOPS-20 System Manager's Guide (AA-4169F-TM)

TOPS-10/TOPS-20 Batch Reference Manual (AA-H374A-TK)

CHAPTER 1

INTRODUCTION TO TOPS-20 USER UTILITIES

This manual describes utility programs available to any user of the TOPS-20 operating system.

The following utility programs are covered in this manual:

- The MAIL program, which allows you to send messages to other users of the system (Chapter 2)
- The RDMAIL program, which allows you to read messages sent to you via the MAIL program (Chapter 3)
- The FILCOM program, which allows you to compare two ASCII files or two binary files (Chapter 4)
- The CREF program, which produces cross-reference listings of symbols used in MACRO, FORTRAN, and ALGOL programs (Chapter 5)
- The MAKLIB program, which performs various functions on libraries of relocatable object modules (Chapter 6)
- The DUMPER program, which allows you to save files and directories on tape, and restore these files and directories to disk (Chapter 7)
- The PLEASE program, which allows you to communicate with the system operator (Chapter 8).

1.1 INVOKING THE UTILITIES

To invoke these utilities, you should be familiar with the TOPS-20 log-in procedure. Type the name of the program after the TOPS-20 prompt @ and press RETURN. The utility then prompts you for input. Thus, the general format is:

```
@Utility Name RET  
Utility Prompt
```

1.2 TYPING FILE SPECIFICATIONS

Many of the utilities accept file specifications as arguments. There are two forms of file specifications. The MAIL, RDMAIL, and DUMPER utilities accept file specifications in the following format:

```
dev:<dir>name.typ.gen;att;...;att
```


INTRODUCTION TO TOPS-20 USER UTILITIES

where:

| | |
|-------|--|
| dev: | Indicates a device name, a file structure name, or a defined logical name |
| <dir> | Indicates a directory name |
| name | Indicates the filename of a particular file in the directory |
| .typ | Indicates a file type that helps identify the contents of the file |
| .gen | Indicates a generation number that shows the number of times a file has been changed |
| ;att | Indicates a file attribute such as a file protection or an account string. |

If you omit the dev: field of the file specification, the system assumes that you mean your connected structure. When you omit the <dir> field of the file specification, the system assumes that you mean your connected directory. When you omit the .gen field of the file specification, the system assumes that you mean the highest generation (largest generation number) for source files. For Destination files, the system assumes that you mean the highest generation plus one.

You can use recognition on file specifications in this format. You can use wildcards only in the DUMPER program. (Refer to Chapter 7.) For more information on file specifications, refer to the TOPS-20 User's Guide.

The FILCOM, CREF, and MAKLIB utilities accept file specifications in a slightly different format as follows:

dev:name.typ[PPN]

In this form of a file specification, filenames are restricted to six characters. File types are restricted to three characters. You cannot use recognition, and file generation numbers are not allowed. Therefore, the highest generation of a file is always used. You can use wildcards only in the MAKLIB program. (Refer to Chapter 6.)

The PPN is a project-programmer number, which you use instead of a directory name. To find out the PPN associated with a specific directory, give the TRANSLATE command. For example, if you wish to find out the PPN associated with the directory <ADLEY> on PS: you do the following:

```
@TRANSLATE <DIRECTORY> PS:<ADLEY>RET
PS:<ADLEY> <IS> PS:14,3051
@
```

You can avoid using PPN's in file specifications by defining a logical name that represents the directory you wish to access. Do the following procedure:

1. Give the DEFINE command to define a logical name as the directory.
2. Use the logical name in place of the device name and the PPN when you type the file specification.

INTRODUCTION TO TOPS-20 USER UTILITIES

The following is an example of defining a logical name for a directory and using it with the FILCOM program:

```
@ DEFINE (LOGICAL NAME) ADL: (AS) <ADLEY> RET
@ FILCOM RET
* TTY:=ADL:TEST.MAC,ADL:TEST2.MAC RET
```

You can also define logical names to reference long filenames or particular file generation numbers. This is especially useful with the FILCOM program when you wish to compare two similar files with different generation numbers. For example:

```
@ DEFINE (LOGICAL NAME) A: (AS) FOO.BAR.3 RET
@ DEFINE (LOGICAL NAME) B: (AS) FOO.BAR.4 RET
@ FILCOM RET
* TTY:=A:,B: RET
```


CHAPTER 2

THE MAIL PROGRAM

2.1 INTRODUCTION

You can use the MAIL program to send messages to other users of the system. You can send mail to a single user or to a group of users who are either logged in or not logged in.

2.2 RUNNING MAIL

To run MAIL, type MAIL after the TOPS-20 prompt @ and press the RETURN key. The program responds with the To: prompt as follows:

```
@MAIL RET
To:
```

Type the name of the user to whom you are sending the message, and press RETURN.

If you are sending a message to a group of users, type the names, separating them with commas, and press RETURN. For example:

```
To: Adley,Sartini,McElmoyle RET
```

The program then prompts:

```
CC:
```

Now list any secondary recipients of your message. Type the name or names (separated by commas) and press RETURN. If you do not want to send a copy to others, simply press RETURN after the CC: prompt.

If you type an invalid (nonexistent) user name, the program responds with:

```
?Invalid user name
```

MAIL returns with either the To: prompt or the CC: prompt. Type CTRL/H after either prompt. This retrieves only the names up to the error, and you can type any additional valid names.

You cannot send more than one copy of a mail message to a user. If you type a user name more than once after either the To: or the CC: prompt, the program prints a warning message. For example:

```
To: Adley RET
CC: Adley RET
%Duplicate name pursued - ADLEY
```

THE MAIL PROGRAM

The MAIL program continues after it prints the warning message; however, the program removes the duplicate name from the list of users.

The program then prompts with:

Subject:

Type a description of the message and press RETURN. For example:

Subject: Location of weekly writers meeting (RET)

If your description exceeds one line, you cannot continue the description on a second line; you must continue typing when you reach the end of a line. The system automatically continues your description on the second line by responding with a carriage return line feed sequence. When you have completed typing your description, press RETURN. For example:

Subject: Location of weekly meeting and change in software release date. (RET)

NOTE

The system may interpret a character in the Subject: line (such as a question mark) as a special character. To avoid this, precede the character with a CTRL/V.

MAIL then prompts with:

Message (Terminate with ESC or CTRL/Z):

and waits for you to enter your message. Once you have terminated your message by typing ESC or CTRL/Z, the program informs you that it has processed your message:

Processing mail...

No errors.
-DONE-

and returns you to TOPS-20 command level.

If you send a message to a user who is logged in and accepting links and system messages, that user is informed immediately as follows:

[You have a message from SENDER]

If you send a message to a user who is not logged in, that user is informed the next time he logs in:

@LOGIN (USER) ADLEY (PASSWORD) (ACCOUNT) 341 (RET)
Job 54 on TTY33 23-Apr-79 09:46:05
You have a message
@

THE MAIL PROGRAM

If you make an error in sending mail to a user, you receive one of the following messages:

```
[USER NAME] not sent BECAUSE:  
Invalid directory number
```

or

```
Invalid simultaneous access
```

or

```
No such file type (or some other reason related to why the  
recipient's MAIL.TXT file could not be found)
```

or

```
[USER NAME] not sent BECAUSE:  
Disk quota exceeded
```

NOTE

For additional information on these error messages, refer to Section 2.4, MAIL Messages.

You can use a recovery procedure to resend mail after receiving some of the MAIL error messages. This recovery is particularly helpful when your message is long and you do not want to retype it. The procedure is as follows:

1. Undelete the MAIL.CPY file in your logged-in directory. This file contains the message that could not be sent.
2. Rename the MAIL.CPY file; for example, ERROR.TXT. This prevents MAIL from deleting the file a second time during message processing.
3. After the TOPS-20 prompt @, type:

```
@GET SYS:MAIL (RET)
```

4. The system gives the TOPS-20 prompt once again, and you type:

```
@REENTER (RET)
```

5. After you press RETURN, the system prompts:

```
File name of message file:
```

Now type the new file spec of the renamed MAIL.CPY file, and press RETURN:

```
File name of message file: (file spec) (RET)
```

The MAIL program now proceeds as though you had just typed ESC or CTRL/Z after the message.

THE MAIL PROGRAM

2.3 OPTIONS TO THE MAIL PROCEDURE

The procedure in Section 2.2 describes the most common use of MAIL. Options to this basic procedure are as follows:

1. You can use the TALK command as an alternative to the MAIL program to communicate with a user who is logged in. (For more information on the TALK command, refer to the TOPS-20 Commands Reference Manual.)
2. You can use the INFORMATION MAIL command to check on the status of new mail, either your own or that of other users. (For more information on this command, refer to the TOPS-20 Commands Reference Manual.)
3. If you send mail often to a group of users, you can create a file containing these names. Then, instead of typing all the names each time you send a message, you can type the filename, preceded by an @, after the To: prompt or the CC: prompt. For example, if the file NAME.FIL.1 contains the user names ADLEY, CRUGNOLA, LYONS, type:

```
To: @NAME.FIL.1 RET
```

The filename can also be combined with other user names following the prompt. However, the file must follow the list of additional user names. For example:

```
To: Sartini,McElmoyle,@NAME.FIL.1 RET
```

4. You can use the contents of an indirect file as your message or Subject: line text. The indirect file you use in the Subject: line can contain only one line. You cannot type any additional text on this line with the indirect file. To send the contents of an indirect file as mail, type an @ followed by the name of the file, and press RETURN. You cannot type any additional text before or after the indirect file. For example:

```
@MAIL RET
To: CRUGNOLA RET
CC:
Subject: Macro files RET
Message (Terminate with ESC or CTRL/Z):
```

```
@MACRO.CMD.1 RET
```

```
Processing mail...
```

```
No errors.
```

```
--DONE--
```

```
@
```

In this case, you do not terminate the message with ESC or CTRL/Z, because you are using an indirect file as your message. However, you terminate the file spec by pressing RETURN. If you terminate your message with CTRL/Z, MAIL responds with:

```
@MACRO.CMD.1 CTRL/Z
?Not confirmed
```

To recover, type CTRL/H immediately to retrieve the indirect file spec. Then, press RETURN.

THE MAIL PROGRAM

5. Normally, you send mail to other users. However, you can also send mail to any non-files-only directory on PS:. The most common non-files-only directories are PS:<REMARKS> and PS:<SYSTEM>.

PS:<REMARKS> can be used for recording information and problems that the system staff should be aware of. For example, use PS:<REMARKS> to record any system difficulties, hardware/software problems, or other related items. To send a message to this directory, type REMARKS after the To: prompt. For example:

```
@MAIL RET
To: REMARKS RET
CC: RET
Subject: Supplies RET
Message (Terminate with ESC or CTRL/Z):
```

```
Terminals in Room 216 need additional boxes of paper, size
9 7/8 x 11. ESC
```

Processing mail...

No errors.

-DONE-

@

Generally, to send messages to all users of the system, you enable your WHEEL or OPERATOR capabilities and run MAIL. These messages are called Messages-of-the-Day. You can also send Messages-of-the-Day by connecting to the directory PS:<SYSTEM>. However, most systems are not set up to allow users to connect to this directory. The following example shows an enabled user running MAIL to send a Message-of-the-Day.

```
$MAIL RET
To: SYSTEM RET
CC: RET
Subject: System shut-down RET
Message (Terminate with ESC or CTRL/Z):
```

```
The system will be shut down tomorrow at 5 p.m. for
preventive maintenance. ESC
```

Processing mail...

No errors.

-DONE-

\$

If you attempt to send mail to PS:<SYSTEM> and do not enable WHEEL or OPERATOR capabilities, MAIL prints the following error message:

```
Processing mail...SYSTEM not sent BECAUSE:
WHEEL or OPERATOR capability required
```

To resend the mail, you can follow the recovery procedure described in Section 2.2 after you enable WHEEL or OPERATOR capabilities.

THE MAIL PROGRAM

When you send a message to PS:<SYSTEM>, the following message appears on all terminals that are receiving system messages:

[New Message-of-the-Day available]

Users not logged in to the system at the time you send the message automatically receive new Messages-of-the-Day the next time they log in.

6. You can use MAIL to inform yourself that your batch job is completed. Place commands to MAIL in your control file as shown in the following example. Note that a period is used as the reply to the To: prompt. This character replaces a user name and informs MAIL that the message is to be sent to you. For example:

```
@CREATE (FILE) TEST.CTL (RET)
INPUT: TEST.CTL
00100 @FILCOM (RET)
00200 *TEST.FOR=DIFFER,FOR,ADDEM,FOR/A (RET)
00300 @PRINT TEST.FOR (RET)
00400 @MAIL (RET)
00500 *. (RET)
00600 * (RET)
00700 *BATCH JOB IS DONE (RET)
00800 *^Z
00900 (ESC)
*E (RET)
@
```

NOTE

Use of a period in place of your user name when you run MAIL is not a feature unique to batch. You can use it any time in the MAIL program when you wish to specify yourself as a recipient of mail.

2.4 MAIL MESSAGES

The most common MAIL messages, their descriptions, and suggested user responses follow. Fatal errors are preceded by a question mark (?). Warning messages are preceded by a percent sign (%).

%Duplicate name purged - [USER NAME]

Description: You attempted to send a user more than one copy of a mail message.

Suggested User Response: None. MAIL continues automatically, and eliminates the duplicate name.

THE MAIL PROGRAM

?Invalid user name

Description: You typed an invalid (nonexistent) user name as a recipient of your message.

Suggested User Response: Type CTRL/H after either the To: prompt or the CC: prompt to retrieve only the names up to the error. Type any additional valid names.

?MAIL.CPY Failure

Entire file structure full

Description: The public structure (PS:) is full, and therefore MAIL cannot operate. You receive this message immediately after invoking the program.

Suggested User Response: Delete and expunge some files from your logged-in directory, or wait until some space is freed.

?MAILER is not running. Messages not sent.

Description: Your message was not sent because the MAILER program is not functioning.

Suggested User Response: Send a message to the operator with the PLEASE program (refer to Chapter 8) to report that MAILER is not functioning. Then, once MAILER is functioning, use the recovery procedure described in Section 2.2 to resend your message.

?Not confirmed

Description: You did not press RETURN immediately after typing an indirect file spec.

Suggested User Response: TYPE CTRL/H immediately after the error message to retrieve the indirect file spec. Then, press RETURN to confirm the indirect file spec.

?Processing errors occurred. No mail sent.

Description: There is a problem with your MAIL.CPY file; either MAILER cannot find it, or the file is not in correct format.

Suggested User Response: Check to see if there is another program updating MAIL.CPY. If not, contact your Software Specialist or send a Software Performance Report (SPR) to DIGITAL.

SYSTEM not sent BECAUSE:

WHEEL or OPERATOR capability required

Description: You attempted to send mail to PS:<SYSTEM> and did not enable WHEEL or OPERATOR capabilities.

Suggested User Response: You must enable WHEEL or OPERATOR capabilities before sending mail to PS:<SYSTEM>. To resend the mail, you can follow the recovery procedure described in Section 2.2 after you enable WHEEL or OPERATOR capabilities.

THE MAIL PROGRAM

%Too many user names. 100 is maximum.

Description: You specified too many users as recipients of your message. MAIL only allows up to 100 user names as recipients of a message; it sends your message to the first 100 names but ignores all that exceed the first 100.

Suggested User Response: Send your message to the first 100 names. Next, edit your MAIL.CPY file to retrieve the message text. Then, run MAIL to send the message to the additional names, using "@" to send the edited MAIL.CPY file as the message text.

[USER NAME] not sent BECAUSE:
Invalid directory number

Description: You attempted to send mail to a nonexistent user directory.

Suggested User Response: You cannot send mail to a user who does not have a directory.

[USER NAME] not sent BECAUSE:
Invalid simultaneous access

Description: Another user has the receiver's MAIL.TXT file open for writing.

Suggested User Response: Follow the recovery procedure described in Section 2.2 to resend the message.

[USER NAME] not sent BECAUSE:
No such file type (or some other related reason)

Description: The intended receiver of your message has no MAIL.TXT file.

Suggested User Response: Ask the user to create a MAIL.TXT file to receive mail messages.

[USER NAME] not sent BECAUSE:
Disk quota exceeded

Description: The receiver's directory exceeds its working quota.

Suggested User Response: Some files must be deleted from the directory before mail can be received. You can also enable WHEEL or OPERATOR capabilities to ignore the user's quota. If the user is logged in, you can use the TALK command as an alternative.

THE MAIL PROGRAM

2.5 TECHNICAL NOTES

MAIL works with another program called MAILER when it handles messages. When you type a mail message, MAIL creates a file, MAIL.CPY, in your logged-in directory. This file is closed when you complete your message input. At this point, MAIL sends an IPCF (Inter-Process Communication Facility) packet to the MAILER program to inform it that you want to send a message. (For more information on IPCF, refer to the TOPS-20 Monitor Calls Reference Manual.) MAILER processes the message by appending the contents of MAIL.CPY in your logged-in directory to the file MAIL.TXT in the recipient's logged-in directory. Then it sends an IPCF packet back to MAIL, which notifies you of the status of your message (sent or not sent). At this point, the MAIL.CPY file is deleted from your logged-in directory. Figure 2-1 shows the relationship between MAIL and MAILER.

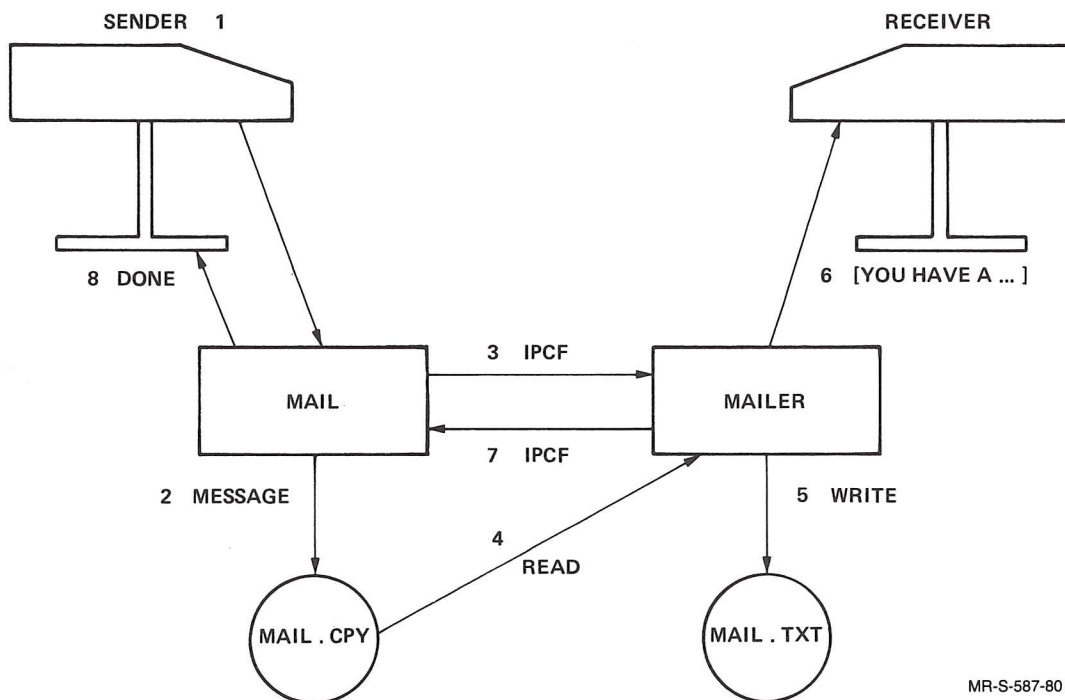


Figure 2-1 The Relationship Between MAIL and MAILER

CHAPTER 3

THE RDMAIL PROGRAM

3.1 INTRODUCTION

The RDMAIL program prints messages that have been sent to you by other users of the system through the MAIL program. Your MAIL.TXT file in your logged-in directory on PS: contains these messages.

NOTE

If your MAIL.TXT file contains mail sent by Version 4 of the MAIL program, you must use Version 4 of RDMAIL to read it.

There are various ways that the system notifies you whenever there is mail that you have not read. If another user sends you mail while you are not logged in, you receive the following message the next time you log in:

```
@LOGIN (USER) DBELL (PASSWORD) (ACCOUNT) 341 RET
Job 35 on TTY42 29-AUG-79 16:14:12
You have a message
@
```

Another user may send you a message while you are logged in. In this case, the system types

```
[You have a message from SENDER]
```

on your terminal.

Messages-of-the-Day sent to you when you are not logged in are printed on your terminal automatically after you log in. However, if your directory is set to REPEAT LOGIN-MESSAGES, you receive all Messages-of-the-Day every time you log in. For more information on the REPEAT LOGIN-MESSAGES subcommand, refer to the BUILD command description in the TOPS-20 Commands Reference Manual. If a system message is sent while you are logged in and you are receiving system messages, you are notified immediately:

```
[New Message-of-the-Day available]
```

THE RDMAIL PROGRAM

You can give the SET MAIL-WATCH command to keep informed of any new mail you receive, especially if you have given the REFUSE SYSTEM-MESSAGES command. (For more information on these two commands, refer to the TOPS-20 Commands Reference Manual.) You can add the SET MAIL-WATCH command to your COMND.CMD file, if you have one, or type it each time you log in. When you give this command, it tells the system to notify you when you have new mail. You receive this notification only when you are at TOPS-20 command level. At intervals of approximately five minutes, the TOPS-20 Command Processor informs you that you have new mail whenever it prompts you for a new command. This message appears on your terminal:

```
[You have new mail]
```

You may give the INFORMATION MAIL command, even if you are not logged in, to check on the status of new mail for yourself or other users. To do this, type the following:

```
@INFORMATION (ABOUT) MAIL (FOR USER) name@RET
```

The system returns with one of the following responses:

```
New mail exists
```

```
or
```

```
No new mail exists
```

```
or
```

```
Mailbox protected
```

3.2 RUNNING RDMAIL

To start RDMAIL, type RDMAIL after the TOPS-20 prompt @ and press the RETURN key. The program responds with the date and time prompt as follows:

```
@RDMAILRET  
Date and time (/HELP for help)
```

If you have enabled WHEEL or OPERATOR capabilities, RDMAIL first asks you whether you want to read your own mail or that of another user. For example:

```
$RDMAILRET  
Special user (y or n)?
```

If you type y, you are indicating that you wish to read another user's mail. Type y and press RETURN. RDMAIL then prompts you to type the name of the user whose mail you wish to read. Type the user name and press RETURN. RDMAIL then prompts you for date and time input. For example:

```
$RDMAILRET  
Special user (y or n)? yRET  
User name: DNeffRET  
Date and time (/HELP for help)
```


THE RDMAIL PROGRAM

If you type n, you are indicating that you wish to read your own mail. Type n and press RETURN. RDMAIL then prompts you for date and time input. For example:

```
$ RDMAIL   
Special user (y or n)?   
Date and time (/HELP for help)
```

RDMAIL allows you to read your messages several ways:

- By giving a date and/or time
- By giving a program switch or combination of switches
- By giving a date and/or time combined with one or more program switches.

To read any new messages, simply press RETURN.

You can define a time period of mail you wish to read. To do this, type a date and/or a time. A common TOPS-20 format is:

MMM DD,YYYY HH:MM:SS

For example, a valid date and time is May 22,1979 17:00:00. If you type only a date, RDMAIL assumes the time 00:01:00. If you type only a time, the program assumes the present date. RDMAIL responds by displaying all messages you received on and after the date and/or time you typed.

If you type an invalid date or time, you receive an error message. Three of the most common ones are:

?Invalid date format

or

?Invalid time format

or

?Day of month too large

After the error message RDMAIL returns with the prompt for you to type a valid date and/or time.

Table 3-1 describes the RDMAIL switches you can use after the prompt, either alone or combined with date/time input.

THE RDMAIL PROGRAM

Table 3-1
RDMAIL Switches

| Switch | Function |
|---------------------|--|
| /ALL | Types all messages, regardless of date. |
| /HELP | Types the program help text, outlining the time/date format and program switches and their functions. |
| /LIST | Outputs messages to the line printer, rather than to your terminal. |
| /MESSAGE-OF-THE-DAY | Types messages from the system Message-of-the-Day file, rather than from your own message file. |
| /PERUSE | Allows you to peruse messages, and gives only the following information for each message: Date: From: To: CC: and Subject. |
| /STOP | Instructs RDMAIL to stop after each message it types. At the end of each message, the system prompts you to type RETURN for more output. |

3.2.1 Reading Mail Using RDMAIL Switches

You type RDMAIL switches immediately following the prompt, and may or may not combine them with a date and/or time. You have the options of combining switches and preceding each switch with a space. To use RDMAIL switches, type a slash (/) followed by the switch name.

/HELP - HELP Switch

Type /HELP to get information on running RDMAIL. For example:

@RDMAIL RET

Date and time (/HELP for help) /HELP RET

After the help text prints on your terminal, the system returns with the prompt for you to type date/time information and/or another switch.

NOTE

/HELP overrides all other switches that you may combine with it. The system ignores all other specified switches in the combination, and prints the full RDMAIL help text.

THE RDMAIL PROGRAM

/ALL - ALL Switch

Type /ALL when you wish to read all messages in the mail file, regardless of date.

/ALL may be combined with all other program switches except /HELP. If you type /ALL after the prompt,

```
@RDMAIL RET
Date and time (/HELP for help) /ALL RET
```

RDMAIL accesses all messages in your file.

/LIST - LIST Switch

Type /LIST when you want messages output to the line printer rather than to your terminal. /LIST can be combined with date/time input, and/or with /ALL. For example,

```
@RDMAIL RET
Date and time (/HELP for help) May 13, 1979 12:00:00 /LIST RET
```

Prints all messages in your file on and after 12:00:00 of May 13, 1979 on the line printer.

If you type only /ALL /LIST after the prompt, RDMAIL prints all messages in your file on the line printer.

/MESSAGE-OF-THE-DAY - System Message Switch

Type /MESSAGE-OF-THE-DAY to print mail from the system Message-of-the-Day file (PS:<SYSTEM>MAIL.TXT), rather than from your own message file. Since new entries in the Message-of-the-Day file are typed on your terminal when you log in, you normally use this switch when a new Message-of-the-Day becomes available while you are logged in. /MESSAGE-OF-THE-DAY may be combined with date/time input. It may also be combined with all other program switches except /HELP. For example:

[New Message-of-the-Day available]

```
@RDMAIL RET
Date and time (/HELP for help) /MESSAGE-OF-THE-DAY RET
```

```
-----
Date: 25 Jun 79 0853-EDT
From: OPERATOR
To: SYSTEM
Subject: STAND-ALONE AT NOON
```

SYSTEM IS GOING DOWN AT NOON FOR NEW MONITOR TO BE LOADED.

```
=====
@
```

In this case, the system prints the Messages-of-the-Day since June 25, 1979 on your terminal.

When you type /MESSAGE-OF-THE-DAY after the prompt, RDMAIL outputs all new Messages-of-the-Day since you last logged in, whether or not you have read them.

THE RDMAIL PROGRAM

/PERUSE - PERUSE Switch

Type /PERUSE when you want to peruse messages in your file. Only the following lines for each message are printed: Date:, From:, To:, CC:, and Subject: /PERUSE can be combined with date/time input and all other program switches except /HELP. A sample output of /PERUSE is as follows:

```
@RDMAIL RET
Date and time (/HELP for help) Apr 12, 1979 /PERUSE RET
-----
Date: 12 Apr 79 0900-EDT
From: OSMAN
To: ADLEY
-----
Subject: your files

-----
Date: 12 Apr 79 1452-EDT
From: HARAMUNDANIS
To: FORADA
CC: ADLEY, HARAMUNDANIS
-----
Subject: DUMPER

-----
Date: 17 Apr 79 1451-EDT
From: LYONS
To: ADLEY
-----
Subject: Re: TEST OF MAIL PROGRAM
```

The system continues to output messages from April 12, 1979 to the present date.

/STOP - STOP Switch

Type /STOP to cause RDMAIL to stop after each message that it types. Following each message, the program prompts you to press RETURN for more output. /STOP can be combined with date/time input and all other program switches except /HELP and /LIST. A sample output of /STOP and /PERUSE is as follows:

```
@RDMAIL RET
Date and time (/HELP for help) May 1, 1979 /STOP /PERUSE RET
-----
Date: 1 May 79 1335-EDT
From: OSMAN
To: ADLEY
-----
Subject: MAILER

[Type <CR> for more] RET
-----
Date: 1 May 79 1844-EDT
From: LYONS
To: ADLEY
-----
Subject: Your account on System 2116

[Type <CR> for more] RET
```

The system continues to output messages from May 1, 1979 to the present date, allows you to peruse them, and stops after each message.

THE RDMAIL PROGRAM

If you type an invalid switch in RDMAIL, you receive the following message:

?Does not match switch or keyword

The system returns with the prompt for you to type a valid switch.

3.3 RDMAIL MESSAGES

The most common RDMAIL messages, their descriptions, and suggested user responses follow. Fatal errors are preceded by a question mark (?). A warning message is preceded by a percent sign (%).

?Day of month too large

Description: You typed an invalid day of the month after the program prompt, Date and time (/HELP for help).

Suggested User Response: Enter a valid day of month when the program returns with the prompt following the error message.

?Does not match switch or keyword

Description: You typed an invalid switch after the program prompt, Date and time (/HELP for help).

Suggested User Response: Type one or more valid RDMAIL switches.

%MAIL.TXT File contains updated entries or improper format

Description: Your MAIL.TXT file is in the wrong format, possibly from being edited with a text editor.

Suggested User Response: Copy the file so you have a record of any current messages. Then, delete the file.

?Invalid date format

Description: You made an error in the date format following the program prompt, Date and time (/HELP for help). This error differs from an invalid day of the month; the error might be, for example, misspelling of the month or an incorrect year.

Suggested User Response: Type a valid date format when the program returns with the prompt after the error message.

?Invalid time format

Description: You typed an invalid time after the program prompt, Date and time (/HELP for help). For example, a possible error is a nonexistent time such as 26:00:00.

Suggested User Response: Type a valid time when the program returns with the prompt after the error message.

THE RDMAIL PROGRAM

?LPT: not available for output

Description: This message appears in response to /LIST. Either you defined a logical name for LPT: that points to an unavailable device. RDMAIL ignores /LIST, and prints the mail on your terminal.

Suggested User Response: Wait for the line printer to become available, or redefine the logical name to point to an available device (such as DSK:).

CHAPTER 4

THE FILCOM PROGRAM

4.1 INTRODUCTION

The FILCOM program compares two files and prints any differences between them. With FILCOM, you can compare either ASCII files (text files and source programs) or binary files (relocatable binary files and save files). The comparison is line by line for ASCII files, and word by word for binary files.

4.2 RUNNING FILCOM

To run FILCOM, type FILCOM, and press the RETURN key. The program prompts you for input with an asterisk:

```
@FILCOM RET
```

*

After the prompt, enter a FILCOM command string in the following format:

Destination file spec=Source file spec1,Source file spec2/Switches

where:

Destination file spec is the output file that contains the differences between the two Source files.

If you do not specify a Destination filename, FILCOM uses the name of the file in Source file spec2. If you omit the name in Source file spec2, the program uses the filename from Source file spec 1. If there is no filename in Source file spec 1, then the filename FILCOM is used. The default for the Destination file type is .SCM for ASCII files and .BCM for binary files. If you completely omit the Destination file spec, FILCOM writes the output to the device TTY:.

Source file spec1 is the first input file you wish to compare.

You must completely specify this file spec in the command string.

Source file spec2 is the second input file you wish to compare.

If you omit the filename in Source file spec2, FILCOM uses the filename in Source file spec1. If you omit the file type in Source file spec2, FILCOM uses the file type in Source file spec1. To indicate a null file type, simply type a period (.) at the end of the filename in either Source file spec1 or Source file spec2.

THE FILCOM PROGRAM

NOTE

FILCOM does not accept file generation numbers. You can still compare two files with the same name and type but different generation numbers (for example, FOO.BAR.1 and FOO.BAR.2) by defining logical names for these files. For more information on defining logical names, refer to the TOPS-20 User's Guide.

You can enter switches after the two input file specs. These switches tell FILCOM how to compare the specified files. However, you don't always need to give switches, because FILCOM often determines the type of comparison by the file types. If either of the input files is of a type listed in Table 4-1, the files are compared in binary mode; otherwise they are compared in ASCII mode.

Table 4-1
Special File Types Recognized by FILCOM

| | | | |
|---|------|------|------|
| Pure Binary Files: | .BAC | .DMP | .RIM |
| | .BIN | .MSB | .RMT |
| | .BUG | .OVR | .RTB |
| | .CAL | .QUC | .SFD |
| | .CHN | .QUE | .SYS |
| | .DAE | .QUF | .UFD |
| | .DCR | .REL | .XPN |
| | | | |
| Sharable Save Files: | .EXE | | |
| Nonsharable Save Files: | .LOW | .SAV | .SVE |
| Offset Address Files; word 0 of the file treated as if it was word 400000. | | | |
| | .HGH | .SHR | |

NOTE

If FILCOM cannot determine the mode for comparison from the input file type or switches, it compares the files in ASCII mode.

For more information on sharable and nonsharable save files and the control words used in them, refer to the TOPS-20 Monitor Calls Reference Manual.

THE FILCOM PROGRAM

After you enter the command string specifying the mode for comparison, the two input files, and any necessary switches, press RETURN. When FILCOM has finished the comparison, it notifies you of the status:

%files are different
or
No differences encountered

The program then prints a second asterisk for you to enter another command string. For example:

```
@FILCOM RET
*COMPAR.FIL=EXFILE.1,EXFILE.2 RET
%files are different
```

*

If you wish to stop the program, type CTRL/C to return to TOPS-20 command level.

4.2.1 Comparing ASCII Files

In ASCII mode, FILCOM compares the characters in each line of the two files, always ignoring nulls. Comments and spacing can be selectively ignored, based on the switches you type.

FILCOM contains eight switches that you use in the command string to compare ASCII files.

- /A Instructs FILCOM to compare the two input files in ASCII mode. It treats both files as if they contain ASCII characters, searches the files for text differences, and ignores similar lines. /A is especially useful if the input files are ASCII files but have one of the file types listed in Table 4-1.
- /B Considers blank lines in the comparison. If you do not specify /B in the command string, FILCOM normally ignores blank lines in the two files.
- /C Instructs FILCOM to ignore comments and spacing in the files. Comments are defined as text on a line following a semicolon. Spacing is defined as spaces and tabs. FILCOM normally considers comments and spaces in the comparison. This switch is useful when you compare assembly language source files (MACRO Assembler source files).
- /H Prints a FILCOM help text, which contains a description of the program and all switches, on your terminal. You can type /H by itself immediately after the FILCOM prompt, or in a command string.

NOTE

/H overrides all other switches that you may combine with it. The system ignores all other specified switches in the combination, and prints the full FILCOM help text.

THE FILCOM PROGRAM

- /nL Defines the number of lines that end a match. When FILCOM finds that number of successive lines that are the same in both input files, it prints all differences found up to the time of the match. The FILCOM output includes the first line of the match for easy reference. FILCOM normally uses the value "3" for the number of lines (the value of n).
- /Q Instructs FILCOM to print only the status of the comparison (either ?files are different or No differences encountered). FILCOM does not enumerate the differences between the files. It stops reading the files after it discovers the first difference.
- /S Ignores spaces and tabs in the comparison of two ASCII files. FILCOM normally considers spaces and tabs in the comparison.
- /U Compares your two input files in update mode. This means that FILCOM creates an output file, which is the second input file, with change bars in the left margin next to the lines that differ from those in the first input file. Deleted lines are indicated by a change bar on the next common line. /U is helpful when you are comparing two versions of text. To obtain a meaningful comparison, type the latest version of the file as the second input file in the command string.

The output file in ASCII mode comparison includes a header for each input file that contains the following information:

- the file number
- the file spec
- the date and time the input file was created.

The following is an example of a header that would appear in an ASCII output file:

```
File 1) DSK:FORLIB.TXTL4,244]   created: 1052 26-JUL-1979
File 2) DSK:FORTX.TXTC4,244]   created: 1155 26-JUL-1979
```

NOTE

If you use /U in the FILCOM command string (compare in update mode), this header does not appear.

Each time FILCOM finds differences between two ASCII input files, it outputs a number corresponding to the first file, and the differences. At the end of the list of differences, the program prints a common line between the two files. The program then prints four asterisks, a number corresponding to the second input file, and the differences. Then FILCOM repeats the set of differences until all the differences between the two files are found. A row of 14 asterisks (*) marks the end of a difference.

THE FILCOM PROGRAM

For example, you have two text files named FIL1.TXT and FIL2.TXT. Use the TYPE command to examine the contents of both files:

```
@TYPE FIL1.TXT RET
this is line 1
this is line 2
this is line 3
this is line 4
this is line 5
this is line 5.5
this is line 6
this is line 7
this is line 8
this is line 9  ;this is a comment
this is line 10
@TYPE FIL2.TXT RET
this is line 1
this is line 2
this is line 3
this is line 4
this is line 5.5
this is line 6
this is line 7

this is line 8
this is line 9
this is line 10
this is line 11
this is line 12
this is line 13
this is line 14, which is the end.
@
```

Run FILCOM to compare these files and name the output file DIFFER.SCM. Type the following:

```
@FILCOM RET
*DIFFER.SCM=FIL1.TXT,FIL2.TXT RET
%files are different
*
```

THE FILCOM PROGRAM

The program informs you that the files are different, and then gives you the asterisk prompt for more input. To see the differences that FILCOM found between the two files, return to TOPS-20 command level (type CTRL/C) and use the TYPE command.

```
%files are different
*(CTRL/C)
@TYPE DIFFER.SCM(RET)
FILE 1) DSK:FIL1.TXTC4,67]      created: 1212 25-Jul-1979
File 2) DSK:FIL2.TXTC4,67]      created: 1616 25-Jul-1979

1)1      this is line 5
1)        this is line 5.5
****
2)1      this is line 5.5
*****
1)1      this is line 9 ;this is a comment
1)        this is line 10
****
2)1      this is line 9
2)        this is line 10
2)        this is line 11
2)        this is line 12
2)        this is line 13
2)        this is line 14, which is the end.
*****
@
```

The output shows the differences between the two files. Line 5 was deleted, line 9 was changed, and lines 11-14 were inserted. The two blank lines in FIL2.TXT are ignored, because /B was not specified in the command string. The number "1" that appears beside 1) and 2) in the output is the page number of the file where the differences were found. Text pages are delimited by formfeeds.

Now, compare FIL1.TXT and FIL2.TXT using /C in the command string.

```
@FILCOM(RET)

*TTY:=FIL1.TXT,FIL2.TXT/C(RET)
File 1) DSK:FIL1.TXTC4,67]      created: 1212 25-Jul-1979
File 2) DSK:FIL2.TXTC4,67]      created: 1616 25-Jul-1979

1)1      this is line 5
1)        this is line 5.5
****
2)1      this is line 5.5
*****
2)1      this is line 11
2)1      this is line 12
2)1      this is line 13
2)1      this is line 14, which is the end.

%files are different
*
```

The output is different because /C causes FILCOM to ignore comments and spacing in the files.

THE FILCOM PROGRAM

Using the same two text files, now compare them in update mode, and write the output to the device TTY: (your terminal). Because FIL2.TXT is the latest version of the two text files, type it as the second input file in the command string. The command string is:

```
@FILCOM RET
*TTY:=FIL1.TXT,FIL2.TXT/U RET
      this is line 1
      this is line 2
      this is line 3
      this is line 4
:      this is line 5.5
      this is line 6
      this is line 7
:
:      this is line 8
:      this is line 9
:      this is line 10
:      this is line 11
:      this is line 12
:      this is line 13
:      this is line 14, which is the end.
%files are different

*
```

As mentioned previously, in update mode comparisons, the output file is the second input file (latest version) with change bars inserted next to the differences found between the two files. The example above shows such an output file. Note that the program also types the "%files are different" status on your terminal. Following this, FILCOM gives another prompt for another command string.

4.2.2 Comparing Binary Files

FILCOM automatically determines that a file is binary if it has one of the file types listed in Table 4-1.

Sharable and nonsharable save files represent the location of data in memory. In FILCOM, expanding the files before comparing them means to compare the data as it would appear if the files were loaded into memory. Comparing the files without expanding them means to compare each word in the files regardless of the usual meaning of the files' control words.

You can list a binary file with FILCOM. To do this, simply omit the comma and the second input file spec in the command string.

THE FILCOM PROGRAM

FILCOM contains seven switches that you use in the command string to compare binary files. Switches control what part of the binary file you see.

- /E Forces FILCOM to consider both input files as sharable save files regardless of the file types given. Normally, FILCOM selects its comparison according to the file types of the files.
- /H Prints the FILCOM help text. Refer to the description of /H in Section 4.2.1.
- /nL Compares a binary file starting at word "n". The number "n" is an octal number. Refer to /nU, below.
- /Q Instructs FILCOM to print only the status of the comparison. It does not list the actual differences, and causes FILCOM to stop reading the files after it discovers the first difference. Refer to the description of this switch in Section 4.2.1.
- /nU Compares a binary file up through word "n". The value "n" is an octal number as in /nL. If you combine /nU with /nL in the command string, the input files are compared only within these limits.
- /W Compares two binary files that have nonstandard file types. (Refer to Table 4-1.) The files are not expanded before FILCOM compares them. /W compares the files' internal control words in addition to data, reading the files one word at a time.
- /X Instructs FILCOM to expand nonsharable save files before comparing them in binary mode. The program ignores control words and compares only the code in the files.

All FILCOM binary switches apply when you dump a file. Expanding a file shows how it would appear in memory. Dumping a file without expanding it shows the file's internal format.

The output file of a binary mode comparison contains the same header as output files for ASCII comparisons. (Refer to Section 4.2.1.) However, the comparison differs because it is done word by word. If the left halves of the two words being compared are the same, FILCOM prints the absolute difference between the two words. Otherwise, FILCOM prints the logical exclusive OR (XOR).

THE FILCOM PROGRAM

For example, you want to compare two binary files. They are FIL3.BIN and FIL4.BIN. First you use FILCOM to dump them and examine their contents. The output is written to the device TTY:(your terminal):

```
@FILCOM RET
```

```
*TTY:=FIL3.BIN RET
```

```
000000 201400 000000
000001 202400 000000
000002 202600 000000
000003 203400 000000
000004 203500 000000
000005 203600 000000
000006 203700 000000
000007 204400 000000
000010 204440 000000
000011 204500 000000
```

```
%files are different
```

```
*TTY:=FIL4.BIN RET
```

```
000000 201400 000000
000001 202400 000000
000002 202600 000000
000003 203400 000000
000004 203540 000000
000005 203600 000000
000006 203700 000000
000007 204420 000000
000010 204440 000000
000011 204500 000000
```

```
%files are different
```

*

Now compare the files. Note that since both file types are .BIN, the program automatically compares them in binary mode without you specifying any switches. Again, the output is written to your terminal:

```
@FILCOM RET
```

```
*TTY:=FIL3.BIN,FIL4.BIN RET
```

```
File 1) DSK:FIL3.BIN[4,67]
File 2) DSK:FIL4.BIN[4,67]
```

```
created: 1419 10-Sep-1979
created: 1419 10-Sep-1979
```

```
000004 203500 000000 203540 000000 000040 000000
000007 204400 000000 204420 000000 000020 000000
```

```
%files are different
```

*

Compare the files once more, but specify a quick comparison (/Q) in the command string. This causes FILCOM to merely report the status of the comparison. If there are differences, the status message prints

THE FILCOM PROGRAM

with a question mark, ?, instead of a percent sign, %, for error detection jobs. (Refer to Section 4.4, FILCOM Messages.) The command string for this comparison is as follows:

```
@FILCOM RET
```

```
*TTY:=FIL3.BIN,FIL4.BIN/Q RET
```

```
File 1) DSK:FIL3.BIN[4,67]
```

```
created: 1419 10-Sep-1979
```

```
File 2) DSK:FIL4.BIN[4,67]
```

```
created: 1419 10-Sep-1979
```

```
?files are different
```

```
*
```

Do a third comparison of these two files, but now restrict the range with /nU and /nL:

```
@FILCOM RET
```

```
*TTY:=FIL3.BIN,FIL4.BIN/5L/6U RET
```

```
No differences encountered
```

In the following example, you are comparing two executable files. FILCOM automatically expands them because of the .EXE file type:

```
@FILCOM RET
```

```
*TTY:=FIL1.EXE,FIL2.EXE RET
```

```
File 1) DSK:FIL1.EXE[4,67]
```

```
created: 1426 10-Sep-1979
```

```
File 2) DSK:FIL2.EXE[4,67]
```

```
created: 1427 10-Sep-1979
```

```
000141 600000 000000 255000 000000 455000 000000
```

```
002112 000004 015062 000004 015063 000001
```

```
%files are different
```

```
*
```

In this last example, you compare both .EXE files without expanding them first. Any differences that exist in the files' internal directory pages appear in the output. The command string for this comparison is as follows:

```
@FILCOM RET
```

```
*TTY:=FIL1.EXE,FIL2.EXE/W RET
```

```
File 1) DSK:FIL1.EXE[4,67]
```

```
created: 1426 10-Sep-1979
```

```
File 2) DSK:FIL2.EXE[4,67]
```

```
created: 1427 10-Sep-1979
```

```
000000 001776 000003 001776 000007 000004
```

```
000002 002000 000000 000000 000000 002000 000000
```

```
000003 001775 000003 000000 000002 001775 000001
```

```
000004 000000 254000 000000 000001 253777
```

```
000005 000000 000140 300000 000003 300000 000143
```

```
000006 001777 000001 000000 000002 001777 000003
```

```
000007 000000 000000 001775 000003 001775 000003
```

```
000010 000000 000000 000000 254000 254000
```

```
000011 000000 000000 000000 000140 000140
```

```
000012 000000 000000 001777 000001 001777 000001
```

```
001141 600000 000000 255000 000000 455000 000000
```

```
003112 000004 015062 000004 015063 000001
```

```
%files are different
```

```
*
```


THE FILCOM PROGRAM

4.3 FILCOM SWITCHES

Table 4-2 contains all the FILCOM switches in alphabetical order. It also lists the mode of comparison and description for each switch.

Table 4-2
FILCOM Switches

| Switch | Comparison | Description |
|--------|---------------|---|
| /A | ASCII | Compares files in ASCII mode |
| /B | ASCII | Considers blank lines in the comparison |
| /C | ASCII | Ignores comments and spacing in MACRO source files |
| /E | Binary | Considers both files as sharable save files |
| /H | ASCII& Binary | Prints the FILCOM help text |
| /nL | ASCII& Binary | Defines the number of lines that end a match in ASCII comparisons; determines the lower limit where a partial comparison begins in binary comparisons |
| /nU | Binary | Determines the upper limit where a partial binary comparison stops |
| /Q | ASCII& Binary | Prints only the status of the comparison; does not enumerate the differences |
| /S | ASCII | Ignores spacing and tabs in the comparison |
| /U | ASCII | Compares the files in update mode and inserts change bars next to the differences |
| /W | Binary | Compares binary files with nonstandard file types, ignoring control words (if any) |
| /X | Binary | Expands nonsharable save files before comparing them |

THE FILCOM PROGRAM

4.4 FILCOM MESSAGES

Some of the messages printed by FILCOM contain information that is dependent on the exact command string, switch, or file you specified. The key to these message variables follows:

| | |
|----------|---|
| [device] | A device name. |
| [file] | A file spec. |
| [n] | A designator for the first or second input file. |
| [reason] | The reason for a file access failure. These are listed in Table 4-3 at the end of this section. |

The most common FILCOM messages are listed below alphabetically. They are all fatal errors because they are preceded by a question mark (?).

?Buffer capacity exceeded and no core available

Description: You attempted to compare two text files with a difference so large that FILCOM cannot obtain enough memory to store the differences.

Suggested User Response: Check that you are comparing two files that are reasonably similar; or, use /nL with n larger than the value 3.

?Command Error

Description: You typed an invalid command, such as an undefined switch or an improperly formatted file spec.

Suggested User Response: Retype the correct command syntax.

?Device [device] Not available

Description: The device you specified is not available. In other words, someone may be using it, or there may be no such device on your system.

Suggested User Response: Specify a device available to you.

?FILE [n] NOT IN CORRECT EXE FORMAT

Description: The first or second input file in the command string is not a correctly formatted sharable save file (.EXE file type). You may have specified a file that is in nonsharable save file format (a result of the TOPS-20 CSAVE command, rather than the SAVE command).

Suggested User Response: Do not use nonsharable save files. They are less efficient than regular sharable save files. Bring the program into memory with the TOPS-20 GET command, save it again with the SAVE command, and then reissue the commands. You can also look at the files as nonsharable save files or pure binary files.

?File [n] not in SAV format

Description: Your first or second input file is not in proper nonsharable save file format.

Suggested User Response: Specify the correct file or look at the files as sharable save files or pure binary files.

THE FILCOM PROGRAM

?File [n] read error

Description: Your first or second input file contains an error.

Suggested User Response: Try the operation again. If it still fails, ask the operator to check the device for errors.

?Input error for input file [n]- [file] [reason]

Description: FILCOM could not access your first or second input file for the reason specified. (Refer to Table 4-3 for the exact reason.)

Suggested User Response: The particular reason in Table 4-3 gives corrective action.

?NOT ENOUGH CORE AVAILABLE TO HOLD DIRECTORY

Description: FILCOM is attempting to compare your sharable save files, but cannot get enough memory to hold the file's internal directory.

Suggested User Response: Re-create the .EXE file. If the error persists, contact your Software Specialist, or send a Software Performance Report (SPR) to DIGITAL.

? Output device error

Description: FILCOM received an error while writing your file.

Suggested User Response: The device may be faulty. If the problem persists, contact your operator to fix the problem.

?Output device error- [device] cannot do output

Description: You specified a device that is unable to do output, such as a card reader.

Suggested User Response: Specify a device that is capable of producing output.

?Output ENTER error for [file] [reason]

Description: FILCOM is unable to write the file for the reason specified.

Suggested User Response: Refer to Table 4-3 for corrective action that applies to the reason for the error.

Table 4-3 contains the various reasons for file access errors that can appear in FILCOM messages.

THE FILCOM PROGRAM

Table 4-3
Reasons for File Access Errors

(0) File not found

You specified a file that does not exist. Specify an existing file.

(1) Nonexistent UFD

You specified a directory that does not exist. Specify an existing directory.

(2) Protection failure

You do not have sufficient privileges to access the named file. Negotiate the needed privileges with the system operator or the owner of the file.

(3) File being modified

Another job is currently modifying the named file. Try to access the file at another time or use a different filename.

(14) No room or quota exceeded

You exceeded the quota of the named directory or the entire capacity of the named file structure. Delete and expunge some of your files, or specify a directory or structure with sufficient space.

(15) Write-lock error

You requested an output file on a write-locked device, such as a magnetic tape. Either write-enable the device and try again, or specify another device.

CHAPTER 5

THE CREF PROGRAM

5.1 INTRODUCTION

The CREF program generates cross-reference listings from .CRF files. You produce .CRF files with LOAD-class commands. (Refer to the TOPS-20 Commands Reference Manual.) A cross-reference listing is one that contains your source program (from either the ALGOL or FORTRAN compilers, or the MACRO assembler) plus a list of all the symbols used and the line numbers on which they are used. As such, the CREF program is a helpful tool for debugging and modifying programs written in these languages.

NOTE

You do not need to run CREF for COBOL programs. The COBOL compiler automatically generates CREF listings.

5.2 RUNNING CREF

Because CREF reads only .CRF format files, you must first create these files before you run CREF. These files are created when your program is compiled. Section 5.2.1 describes how to create .CRF files.

Once you create .CRF files, you can run CREF to produce cross-reference listings. Section 5.2.2 describes how to run CREF.

5.2.1 Creating .CRF Files with COMPILE

To create a .CRF file, you must compile your program. To do this, type COMPILE after the TOPS-20 prompt @, followed by /CREF and the name of the program you want to compile; then press the RETURN key. After you press RETURN, the system compiles your program and then returns you to TOPS-20 command level. For example:

```
@COMPILE/CREF BE.MAC RET
MACRO: BE

EXIT
@
```


THE CREF PROGRAM

Compiling a program does not automatically create a .CRF file. This is why you specify /CREF in the command string. For more information on the COMPILE command, refer to the TOPS-20 Commands Reference Manual.

You can create .CRF files of two programs in the same command string. For example:

```
@COMPILE/CREF BE.MAC,CREFA.MAC RET
MACRO: BE
MACRO: CREFA

EXIT
@
```

Note that this one command string is equivalent to creating .CRF files of both programs separately, as in the example below:

```
@COMPILE/CREF BE.MAC RET
MACRO: BE

EXIT
@COMPILE/CREF CREFA.MAC RET
MACRO: CREFA

EXIT
@
```

If your program has already been compiled, the above procedure will not work. You will immediately return to TOPS-20 command level after typing the command string. To create a .CRF file of an already compiled program, you must recompile the program. Type the COMPILE command followed by /COMPILE, /CREF, the program name, and press RETURN. For example:

```
@COMPILE/COMPILE/CREF BE.MAC RET
MACRO: BE

EXIT
@
```

5.2.2 Producing Cross-Reference Listings

Once you create a .CRF file, you can run CREF to generate the actual cross-reference listing. There are four types of tables that you can get in these listings. The four types of tables are:

- Regular Symbols - This table contains user-defined symbols, labels, address tags, and assignments.
- OPDEF and Macro Names - This table contains user-defined operators such as macro calls and OPDEFs.
- Op Codes - This table contains hardware machine mnemonics and assembler pseudo-ops, such as MOVE and XALL.
- Procedure Nesting Levels - This table contains the names of procedures and numbers of blocks, indented to show their nesting in the program. Only the ALGOL compiler generates this table in a CREF listing.

THE CREF PROGRAM

You produce cross-reference listings by running CREF and giving a command string with switches to specify the type of listing you desire. The general command string format is:

Destination file spec=Source file spec1/Switches,Source file spec2/Switches...

where:

Destination file spec is the output file that contains the cross-reference listing.

If you omit the device and the filename in the Destination file spec, CREF uses the device LPT:. If you omit the device but do specify a filename, CREF uses the device DSK:. If you do not specify a filename in the Destination file spec, CREF uses the filename in Source file spec1. If you do not specify a file type in the Destination file spec, CREF uses the type .LST. If you omit the equal sign (=) in the command string, CREF uses all defaults for the Destination file spec (you only specify the Source file specs).

Source file spec1,Source file spec2...are the input files. These are the .CRF files you produced from compiling your programs.

If you omit the device in any of the Source file specs, CREF uses the device DSK:. If you omit a filename in any Source file spec, CREF uses the filename CREF. If you omit a file type in any Source file spec, CREF uses the file type .CRF, then .LST, .TMP, and null.

If you omit a PPN or a logical name for a PPN in either the Destination file spec or any of the Source file specs, CREF assumes that you mean your connected directory.

As mentioned, you can type CREF program switches in the command string after each input file spec. These switches and their functions are described in Table 5-1.

THE CREF PROGRAM

Table 5-1
CREF Switch Options

| Switch | Function |
|--------|---|
| /A | Advances magnetic tape reel by one file. You can type this switch more than once in the command string. |
| /B | Backspaces magnetic tape reel by one file. You can type this switch more than once in the command string. |
| /C | Cancels the processing of any switches in your SWITCH.INI file. |
| /D | Restores the processing of any default switches in your SWITCH.INI file. |
| /H | Types the CREF help file. /H is illegal in a SWITCH.INI file. |
| /K | Suppresses Regular Symbol Table in the CREF listing. |
| /M | Suppresses OPDEF/Macro Table in the CREF listing. |
| /O | Includes the Op Code Table in the CREF listing. |
| /P | Preserves an input file with the file type .CRF or .LST. These types of input files are normally deleted. |
| /R | Requests the line number at which the CREF listing is to start. CREF types out "RESTART LISTING AT LINE:", after which you type the line number and press RETURN. If you use an indirect file, CREF looks for the number in the indirect file. /R is most useful for physical (non-spooled) line printers, and is illegal in a SWITCH.INI file. |
| /S | Suppresses the program listing and lists only the tables you select. |
| /T | Skips to the logical end of the magnetic tape. |
| /W | Rewinds the magnetic tape. |
| /Z | Zeroes the DEctape directory. This is a historical switch, and is illegal. |

THE CREF PROGRAM

If you always want to use specific switches when you run CREF, you can put these switches in a SWITCH.INI file. Then, each time you run CREF, it automatically reads these switches from the SWITCH.INI file. You can use all CREF switches in your SWITCH.INI file except /H and /R. For more information on creating a SWITCH.INI file, refer to the TOPS-20 User's Guide.

You can use an indirect file in CREF to reference a file within a command string. The indirect file can contain a complete or partial CREF command string (filenames and switches). For more information on using indirect files, refer to the TOPS-20 User's Guide.

You can produce cross-reference listings by typing CREF as a command or as a program name after the TOPS-20 prompt @. To run CREF as a command, type CREF after the TOPS-20 prompt @ and press RETURN. This causes CREF to produce a cross-reference listing of any .CRF files you created in that particular terminal session. For example, if you create .CRF files for BE.MAC and CREFA.MAC and then run CREF, type the following:

```
@CREF RET
CREF: BE
CREF: CREFA
@
```

Because you did not specify a switch, the default is for CREF to give you a cross-reference listing that contains a Regular Symbols Table and an OPDEF/Macro Table for each of the two .CRF files. If you only specify switches with this format, these switches apply to all files CREF processes. If one of your CRF files is an ALGOL program, the listing also includes the Procedure Nesting Level Table.

If you create .CRF files for several programs, but want a cross-reference listing of just one file, type CREF after the TOPS-20 prompt @ followed by a CREF command string. For example, you have .CRF files for BE.MAC and CREFA.MAC, but you only want a cross-reference listing for BE.MAC. Type the following:

```
@CREF BE.LST=BE.CRF/Switch RET
CREF: BE
@
```

This command string causes CREF to produce a cross-reference listing for BE.MAC that includes any tables you specify with the switch.

To run CREF as a program, type R CREF after the TOPS-20 prompt @ and press RETURN. The program prompts you with an asterisk. Now type a CREF command string. For example, you create a .CRF file for BE.MAC and then run CREF as a program, specifying /O in the command string. Type the following:

```
@R CREF RET
*BE.LST=BE/O RET
[CRFXKC 5K CORE]
*
```

This command string causes CREF to produce an cross-reference listing for BE.MAC that contains a Regular Symbols Table, an OPDEF/Macro Table, and the Op Code Table (/O appears in the command string).

THE CREF PROGRAM

NOTE

For an explanation of the program message in square brackets that appears in the last example, refer to Section 5.5, CREF Messages.

You can also run other programs from CREF. To do this, type R CREF after the TOPS-20 prompt @ and press RETURN. After the asterisk prompt, type the filespec for the second program you wish to run, then type an exclamation point (!) and press RETURN.

5.3 CREF EXAMPLES

First, type your SWITCH.INI file to see that /O is included for CREF. Then define the logical name LPT: so that the CREF output goes to the disk, where you can type it. Without the logical name, the output is spooled to the line printer.

```
@ TYPE SWITCH.INI (RET)
CREF/O
RUNOFF/CRETURN/MESSAGE:(NOPREFIX,FIRST)
MAKLIB/MESSAGE:(NOPREFIX,FIRST)
@ DEFINE LPT: DSK: (RET)
@
```

Now, compile a MACRO assembler program and request a CREF listing. Process MACRO's listing file with CREF to get the final listing. Note that because of the /O on the CREF line in SWITCH.INI, CREF automatically includes the Opcode Table.

THE CREF PROGRAM

@COMPILE/COMPILE/CREF CREFA RET

MACRO: CREFA

EXIT

@CREF RET

CREF: CREFA

@TYPE CREFA.LST RET

LCREFA CREF Example MACRO %53A(1152) 10:39 10-Sep-79 Page 1

CREFA MAC 11-Jul-79 11:54 Show What CREF Does

| | | | | | | |
|----|---------|--------------------|--|----------|---------------------|--------------|
| 1 | | | | TITLE | CREFA | CREF Example |
| 2 | | | | SUBTTL | Show What CREF Does | |
| 3 | | | | | | |
| 4 | | | | SEARCH | MACSYM, MONSYM | |
| 5 | | | | SALL | | |
| 6 | | | | NOSYM | | |
| 7 | | | | | | |
| 8 | | 000000 | | T0=0 | | |
| 9 | | 000016 | | L=16 | | |
| 10 | | 000017 | | P=17 | | |
| 11 | | | | | | |
| 12 | | 000017 | | CONST=17 | | |
| 13 | | | | ENTRY | TIMES. | |
| 14 | | | | | | |
| 15 | 000000' | 201 00 0 00 000017 | | TIMES.: | MOVX | T0,CONST |
| 16 | 000001' | 220 00 1 16 000000 | | | IMUL | T0,@(L) |
| 17 | 000002' | 263 17 0 00 000000 | | | RET | |
| 18 | | | | | | |
| 19 | | | | END | | |

NO ERRORS DETECTED

PROGRAM BREAK IS 000003

CPU TIME USED 00:00.872

67P CORE USED

| | | | |
|---------|-----|-----|----|
| ^LCONST | 12# | 15 | |
| L | 9# | 16 | |
| P | 10# | | |
| T0 | 8# | 15 | 16 |
| TIMES. | 13 | 15# | |
| ..MX1 | 15# | 15 | 16 |
| ..MX2 | 15# | 16 | |
| .PSECT | 15 | 16 | |
| ^LMOVX | 15 | | |
| RET | 17 | | |
| ^LEND | 19 | | |
| ENTRY | 13 | | |
| IFDEF | 15 | | |
| IFE | 15 | 16 | |
| IFNDEF | 16 | | |
| IMUL | 16 | | |
| MOVEI | 15 | | |
| NOSYM | 6 | | |
| PURGE | 16 | | |
| SALL | 5 | | |
| SEARCH | 4 | | |
| SUBTTL | 2 | | |
| TITLE | 1 | | |
| .IF | 15 | | |
| .IFN | 15 | | |
| .PSECT | 15 | 16 | |
| @ | | | |

THE CREF PROGRAM

Now, compile a FORTRAN program. Generate the CREF listing and type the result.

@COMPILE/COMPILE/CREF CREFB RET

FORTRAN: CREFB

TIMES

@CREF RET

CREF:

@TYPE CREFB.LST RET

^LTIMES

CREFB.FOR

FORTRAN V.5A(621) /KI/C 10-SEP-79

10:39

PAGE 1

```
00001      FUNCTION TIMES(ARG)
00002
00003      PARAMETER CONST = 17
00004      INTEGER TIMES, ARG
00005
00006      10  TIMES = CONST*ARG
00007      RETURN
00008
00009      END
```

SUBPROGRAMS CALLED

SCALARS AND ARRAYS ["*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED]

TIMES 1 ARG 2

TEMPORARIES

```
^LARG      1#      4#      6
CONST      3#      6
TIMES      1#      4#      6#
10P        6#
```

TIMES [NO ERRORS DETECTED]

@

THE CREF PROGRAM

Finally, compile an ALGOL program. Generate the CREF listing as before and type the result. Note the Procedure Nesting Table.

@ COMPILE/COMPILE/CREF CREFC RET
ALGOL: CREFC

EXIT

@ CREF RET

CREF:

@ TYPE CREFC.LST RET

DECsystem 10 ALGOL-60, Version 6A(654) 10-SEP-79 10:39:35

Command string: CREFC,CREFC/C=MISC:CREFC.ALG

| | | | | |
|----|--------|----|----|---|
| 1 | 000005 | B1 | 1 | BEGIN REAL X, Y; |
| 2 | 000005 | | 2 | |
| 3 | 000012 | | 3 | REAL PROCEDURE SQUAREROOT(X,L); |
| 4 | 000012 | | 4 | |
| 5 | 000016 | | 5 | VALUE X; REAL X; LABEL L; |
| 6 | 000016 | | 6 | |
| 7 | 000021 | B2 | 7 | BEGIN REAL Y, Z; |
| 8 | 000025 | | 8 | IF X < 0 THEN GOTO L; |
| 9 | 000027 | | 9 | Y := (1 + X)/2; |
| 10 | 000043 | | 10 | IT: Z := (X/Y + Y)/2; |
| 11 | 000060 | | 11 | IF ABS(Z - Y) < 1E-6 THEN GOTO OK; |
| 12 | 000063 | | 12 | Y := Z; GOTO IT; |
| 13 | 000074 | | 13 | OK: SQUAREROOT := Z; |
| 14 | 000076 | E2 | 14 | END; |
| 15 | 000076 | | 15 | |
| 16 | 000115 | | 16 | NG: WRITE("Number please: "); BREAK.OUTPUT; |
| 17 | 000117 | | 17 | READ(X); |
| 18 | 000122 | | 18 | Y := SQUAREROOT(X,NG); |
| 19 | 000127 | | 19 | WRITE("The square root of "); |
| 20 | 000132 | | 20 | PRINT(X); |
| 21 | 000135 | | 21 | WRITE(" is "); |
| 22 | 000140 | | 22 | PRINT(Y); |
| 23 | 000143 | | 23 | BREAK.OUTPUT; |
| 24 | 000145 | E1 | 24 | END; |

NO ERRORS

| PROGRAM | | | | | | | | | |
|-------------|-----|--|--|--|--|--|--|--|--|
| LE---- | 1 | | | | | | | | |
| B---- | 1 | | | | | | | | |
| B---- | 2 | | | | | | | | |
| E---- | 2 | | | | | | | | |
| E---- | 1 | | | | | | | | |
| LABS | 11 | | | | | | | | |
| BREAKOUTPUT | | | | | | | | | |
| | 16 | | | | | | | | |
| IT | 10# | | | | | | | | |
| L | 3# | | | | | | | | |
| NG | 16# | | | | | | | | |
| OK | 11 | | | | | | | | |
| PRINT | 20 | | | | | | | | |
| READ | 17 | | | | | | | | |
| SQUAREROOT | | | | | | | | | |
| | 3# | | | | | | | | |
| WRITE | 16 | | | | | | | | |
| X | 1# | | | | | | | | |
| Y | 1# | | | | | | | | |
| Z | 7# | | | | | | | | |
| @ | | | | | | | | | |

THE CREF PROGRAM

5.4 CREF MESSAGES

The messages printed by CREF fall into three general categories: informational messages, warning messages, and fatal errors. Informational messages are enclosed in square brackets [], and merely inform you of CREF's progress in processing your file. Warning messages are preceded by a percent sign (%) and indicate that something unexpected occurred, but that CREF was able to recover. In this case, verify that your output is correct. Fatal errors are preceded by a question mark (?), and indicate an occurrence that CREF could not handle. In this case, CREF aborts its operation. You must fix the problem before reissuing your command string. The % and the ? preceding the message are easily detected in Batch jobs.

Some of the messages contain variable information that is dependent on the exact command string, switch, or file you specified. These variables are as follows:

| | |
|----------|---|
| [access] | An octal code associated with a file access failure. For possible access failures, refer to Table 5-2 at the end of this section. |
| [device] | A device name. |
| [file] | A file spec. |
| [memory] | A memory size, such as 30K. |
| [status] | An octal code associated with a read or write error while processing a file. The meaning of this code is described in Table 5-3 at the end of this section. |
| [switch] | A switch name. |

The most common CREF messages are listed below alphabetically.

?CRFCDN Can't get command file device [device]

Description: CREF cannot access the named device in your indirect file command.

Suggested User Response: Specify an existing or available device.

?CRFCEF Cannot ENTER file, [access] [file]

Description: CREF could not create the specified file for the reason associated with the printed access code. (Refer to Table 5-2.)

Suggested User Response: Refer to Table 5-2 for corrective action that applies to the reason for the error.

?CRFCFE Command file INPUT error, [status] [file]

Description: An input error occurred reading the named command file. For the meaning of the specified status code, refer to Table 5-3.

Suggested User Response: Refer to Table 5-3 for corrective action that applies to the reason for the error.

THE CREF PROGRAM

?CRFCFF Cannot find file, [access] [file]

Description: CREF cannot find the specified file for the reason associated with the printed access code. (Refer to Table 5-2.)

Suggested User Response: Refer to Table 5-2 for corrective action that applies to the reason for the error.

?CRFCFI Cannot find input file, [file]

Description: CREF cannot find the specified file. The reason can be any of those listed in Table 5-2.

Suggested User Response: Refer to Table 5-2 for corrective action that applies to the reason for the error.

?CRFCLC Can't LOOKUP command file [access] [file]

Description: CREF cannot find the specified command file for the reason associated with the printed access code. (Refer to Table 5-2.)

Suggested User Response: Refer to Table 5-2 for corrective action that applies to the reason for the error.

?CRFCME Command error - type /H for help

Description: You typed an invalid command.

Suggested User Response: Retype the correct command or type /H for help.

?CRFDNA Device not available, [file]

Description: You specified a device in the file specification that is not available to your job. It may be in use by another job.

Suggested User Response: Specify a device that is available to your job.

?CRFIBP Input buffer size phase error - 0 [file]

Description: The size of the buffer area set up by the system is larger than the size expected by CREF.

Suggested User Response: This is an internal error, and not expected to happen. If it does, contact your Software Specialist or send a Software Performance Report (SPR) to DIGITAL.

%CRFIDC Improper input data at line [decimal], continuing

Description: CREF detected incorrect data in your input file. This is most likely a problem with the compiler that generated the CREF input file.

Suggested User Response: First, be sure that you are processing a valid CREF input file. If so, this is an internal error, and not expected to happen. Try to re-create the .CRF file, since the data in the original one may have been corrupted. If the problem persists, contact your Software Specialist or send a Software Performance Report (SPR) to DIGITAL.

THE CREF PROGRAM

?CRFIMA Insufficient memory available

Description: Your input file is too large for CREF to handle.

Suggested User Response: Divide your program into smaller files and try again.

?CRFINE INPUT error, [status] [file]

Description: An input error occurred reading the named file. For the meaning of the specified status code, refer to Table 5-3.

Suggested User Response: Refer to Table 5-3 for corrective action that applies to the reason for the error.

?CRFOUE OUTPUT error, [status] [file]

Description: An output error occurred writing the named file. For the meaning of the specified status code, refer to Table 5-3.

Suggested User Response: Refer to Table 5-3 for corrective action that applies to the reason for the error.

%CRFPUE Please use "=" rather than ""

Description: Older versions of CREF allowed only the use of an underbar ("") to separate the input files from the output file. CREF currently allows either "=" or "", but it is preferable for you to use "=". CREF acts as if you typed "=".

Suggested User Response: Use "=" in the future.

%CRFRLI Restart listing at line:

Description: You specified /R to continue a listing. CREF is requesting the line number of the line to resume printing.

Suggested User Response: Type the desired line number.

%CRFSIH "/H" or "/R" switch illegal in SWITCH.INI defaulting

Description: You specified /H or /R illegally in the CREF line in your SWITCH.INI file. CREF ignores these switches.

Suggested User Response: Remove /H or /R from your SWITCH.INI file.

%CRFSII Syntax error in SWITCH.INI defaults

Description: The CREF commands in your SWITCH.INI file are invalid. CREF ignores them.

Suggested User Response: Correct the invalid switches in your SWITCH.INI file.

?CRFSIO I/O error while reading SWITCH.INI

Description: An input error occurred reading SWITCH.INI switch defaults. The reason for the failure can be any of those listed in Table 5-3.

Suggested User Response: Refer to Table 5-3 for corrective action that applies to the reason for the error.

THE CREF PROGRAM

?CRFUKE Unknown switch "[switch]" {in DSK:SWITCH.INI}

Description: You specified a nonexistent switch.

Suggested User Response: Respecify the command string with a valid switch, or fix your SWITCH.INI file.

[CRFXKC [memory] core]

Description: CREF finished its processing, and is informing you how much memory it used to process your CREF file.

The octal codes listed in Table 5-2 appear in various CREF error messages, wherever [access] is used. The explanation beside each octal code describes the reason for the failure. Appropriate corrective action for each failure follows the explanation.

Table 5-2
Reasons for File Access Errors

| Octal Code | Explanation/Action |
|------------|---|
| 0 | File not found. The named file was not found by CREF. Specify an existing file. |
| 1 | You specified a directory that does not exist. Specify an existing directory. |
| 2 | Protection failure. You do not have sufficient privileges to access the named file. Negotiate the needed privileges with the system operator or the owner of the file. |
| 3 | File being modified. Another job is currently modifying the named file. Try to access the file at another time or use a different filename. |
| 14 | No room or quota exceeded. You exceeded the quota of the named directory, or the entire capacity of the file structure. Delete and expunge some of your files, or specify a directory or structure with sufficient space. |
| 15 | Write-lock error. You requested an output file on a write-locked device, such as a magnetic tape. Either write-enable the device and try again, or specify another device. |

The status code in various CREF error messages is an octal number that is best interpreted as bits. For example, if a CREF message prints the status code 40001, this means that you have exceeded your disk quota. The status code printed by CREF is the logical "OR" of all applicable bit values. Table 5-3 contains these status codes, their meanings, and the corresponding remedial action.

THE CREF PROGRAM

Table 5-3
Error Status Codes

| Status Code | Meaning/Action |
|-------------|--|
| 400000 | The device is write-locked. Either specify a write-enabled device or ask the system operator to write-enable your device. |
| 200000 | A hardware error occurred. The hardware is in error and should be fixed if the problem persists. |
| 100000 | A parity error occurred. The data on the device is incorrect. Correct the data and try again. |
| 40000 | Quota exceeded or structure full. You either exceeded your directory's quota or the entire capacity of the structure. Delete and expunge some files or specify a structure or directory with sufficient space. |

NOTE

All other bit values (such as 20000 or 10) are simply status bits and do not indicate an error. Only those listed above indicate an error. Any other code simply indicates the status.

5.5 TECHNICAL NOTES

The information in this section is primarily for users who write their own compilers (such as MACRO, ALGOL, or FORTRAN) and users who create .CRF files. It describes the .CRF input file format and the various control characters that you use.

The control characters in Tables 5-4 and 5-5 appear in the CREF input file produced by MACRO, FORTRAN and ALGOL. At the beginning of each line of the listing, CREF input data is enclosed by DELETE B and DELETE C. The symbol or instruction types and the number of their component characters are defined by control characters. The set of control characters defining symbols and instructions is the same as the set defining the number of symbol or instruction characters. A control character's position determines its function. For example, in the input CREF data <DELETE>B^C^CEND<DELETE>c, the <DELETE>B indicates the beginning of the data. The first CTRL/C (^C) defines the instruction END as a pseudo-op code. The second CTRL/C (^C) defines the number of characters in the instruction END as 3, and the <DELETE>C terminates the CREF data.

THE CREF PROGRAM

Symbols referenced in a block-structured language such as ALGOL should assign a unique number to each symbol name. This ensures that symbols with the same name but defined in different blocks have different numbers. Then, the number for each symbol should be referenced in CREF data rather than the name. After the last use of a symbol (usually at the end of its block), use CTRL/I to associate the symbol's name with the unique number assigned to it.

Control characters that require two symbols (such as CTRL/I) should have two adjacent length character and symbol name pairs.

The control characters that begin and end the CREF input data are described in Table 5-4.

Table 5-4
Beginning and Ending Control Characters

| Character | ASCII Codes | Meaning |
|----------------------------|---------------------|--|
| A (prints as A) | 177, 101 (Octal) | Terminates the CREF data on each line and adds a horizontal tab to the line of the listing. |
| B (prints as B) | 177, 102 | Signals beginning of the CREF data on each line. |
| C (prints as C) | 177, 103 | Terminates the CREF data on each line. A line number is incremented and printed. This is the line number referenced in the CREF tables. |
| D (prints as D) | 177, 104 | Identical to DELETE A, except that DELETE D does not increment or print line numbers. |
| E (prints as E) | 177, 105 | Causes CREF to print its tables immediately, and reinitializes the tables for another program. FORTRAN uses this function between subroutines. |
| F (prints as F) | 177, 106 | Identical to DELETE C, except that DELETE F does not increment or print line numbers. |

Table 5-5 contains the control characters that define symbols, instruction types, and macros. Except for CTRL/B and CTRL/G, each of these characters precedes the symbol name being referenced.

THE CREF PROGRAM

Table 5-5
Symbol-Definition Control Characters

| Character | ASCII Code | Meaning |
|---------------------------|------------|--|
| <code>(CTRL/A)(^A)</code> | 001 | Declares a reference to a user-defined symbol (such as a regular MACRO symbol or a FORTRAN variable). |
| <code>(CTRL/B)(^B)</code> | 002 | Declares a defining occurrence of a user-defined symbol. This character immediately follows the symbol name. |
| <code>(CTRL/C)(^C)</code> | 003 | Declares a reference to a MACRO pseudo-op or a hardware-defined op code. |
| <code>(CTRL/D)(^D)</code> | 004 | Declares a defining occurrence of a MACRO pseudo-op or a hardware-defined op code. |
| <code>(CTRL/E)(^E)</code> | 005 | Declares a reference to a macro or OPDEF. |
| <code>(CTRL/F)(^F)</code> | 006 | Declares a defining occurrence of a macro or OPDEF. |
| <code>(CTRL/G)(^G)</code> | 007 | Causes CREF to delete the last symbol that it read. |
| <code>(CTRL/H)(^H)</code> | 010 | Combines two symbols that are defined at different block levels and are then found to be the same. The second symbol specified becomes the name for both. For example, in a one-pass block-structured compiler that allows symbol references before their definition, a symbol referenced in an inner block must be treated as a unique symbol until the end of the block. If no local definition is found, this symbol's references must then be combined with an outer block's references. |
| <code>(CTRL/I)(^I)</code> | 011 | Declares the user-defined symbol by giving it a name in place of the unique numeric. The numeric is the first argument, and its name is the second argument. |

THE CREF PROGRAM

Table 5-5 (Cont.)
Symbol-Definition Control Characters

| Character | ASCII Code | Meaning |
|---------------------------|------------|--|
| <code>(CTRL/J)(^J)</code> | 012 | Illegal in CREF as a symbol definition control character. |
| <code>(CTRL/K)(^K)</code> | 013 | Identical to CTRL/I, except that it operates on macros rather than on symbols. |
| <code>(CTRL/L)(^L)</code> | 014 | Illegal in CREF. |
| <code>(CTRL/M)(^M)</code> | 015 | Declares the beginning of a symbol block. The argument is the block name. |
| <code>(CTRL/N)(^N)</code> | 016 | Declares the end of a symbol block. The argument is the block name. |
| <code>(CTRL/O)(^O)</code> | 017 | Declares a line number to use in place of the line's actual position in the file. The line number is specified after the CTRL/O in the same format as other symbols. |

The octal value of each character described in Table 5-6 is used by CREF to determine the number of characters in a symbol name. The same set of characters defines the symbol as well as its number of characters. The character's position determines its function. The character-count character immediately precedes the symbol with no intervening spaces or characters (^CEND, for example). Table 5-6 contains the characters and their meanings.

Table 5-6
Character-Count-Definition Characters

| Character | ASCII Code | Meaning |
|---------------------------|-------------|------------------------------------|
| <code>(CTRL/A)(^A)</code> | 001 (octal) | The symbol contains 1 character. |
| <code>(CTRL/B)(^B)</code> | 002 | The symbol contains 2 characters. |
| <code>(CTRL/C)(^C)</code> | 003 | The symbol contains 3 characters. |
| : | : | : |
| : | : | : |
| : | : | : |
| ? | 077 | The symbol contains 63 characters. |

THE CREF PROGRAM

The following example illustrates the symbols and references recognized by CREF that you can make in an Assembly Language Program. The control characters show the references that are made to particular symbols.

Source File CREFD.MAC:

```
TITLE  CREFD  CREF Example
SUBTTL  Show Internal CREF Information

SEARCH  MACSYM, MONSYM  ;Pseudo-op reference

CONST=17                ;Symbol definition

DEFINE  MACRO(X)<        ;;Pseudo-op reference, macro definition (in-line)
OPCODE  X                ;OPDEF and symbol reference (when expanded)
>

OPDEF   OPCODE|MOVE|J   ;Pseudo-op reference, OPDEF definition and
                        ; opcode reference

START:  MACRO  CONST     ;Symbol and macro reference

END     START            ;Pseudo-op and symbol reference
```

THE CREF PROGRAM

.CRF File CREFD.CRF:

NOTE

In this printout of a .CRF File, DELETE characters are designated as follows:

DELETE R = "PB

| LCREFD CREF Example | MACRO X53A(1152) | 13:44 | 10-Sep-79 | Page 1 |
|--|------------------|----------------|---|---------------------------------|
| CREFD MAC | 10-Sep-79 | 13:44 | Show Internal CREF Information | |
| ~?B~C~ETITLE~?C | TITLE | CREFD | CREF Example | |
| ~?B~C~FSUBTTL~?C | SUBTTL | Show Internal | CREF Information | |
| ~?B~?C | | | | |
| ~?B~C~FSEARCH~?C | SEARCH | MACSYM, MONSYM | ;Pseudo-op reference | |
| ~?B~?C | | | | |
| ~?B~A~ECONST~B~?C | CONST=17 | | ;Symbol definition | |
| ~?B~?C | | | | |
| ~?B~C~FDEFINE~F~EMACRO~?C | DEFINE | MACRO(X)< | | |
| ~?B~C~Fpseudo-op reference, macro definition (in-line) | OPCODE | X | ;OPDEF and symbol reference (when expanded) | |
| ~?B~?C | | | | |
| ~?B~?C | | | | |
| ~?B~C~EOPDEF~C~EMOVEIT~F~FOPCODE~?C | 201000 | 000000 | OPDEF | OPCODE[MOVEIT] |
| ~?B~C~Epseudo-op reference, OPDEF definition and | | | ; opcode reference | |
| ~?B~?C | | | | |
| ~?B~A~ESTART~B~E~EMACRO~?C | START: | MACRO | CONST | |
| ~?B~C~Fmbol and macro reference | | | | |
| ~?B~E~FOPCODE~A~ECONST~?C | 201 00 0 00 | 000017 | OPCODE | CONST |
| ~?B~C~FEF and symbol reference (when expanded) | | | | |
| ~?B~?C | | | | |
| ~?B~C~EEND~C~ESTART~?C | 000000, | | END | START |
| | | | | ;Pseudo-op and symbol reference |

NO ERRORS DETECTED

```
PROGRAM BREAK IS 000001
CPU TIME USED 00:00.904
```

```
67P CORE USED
~LCREFD CREF Example MACRO %53A(1152) 13:44 10-Sep-79 Page 5-1
CREFD MAC 10-Sep-79 13:44 SYMBOL TABLE
```

```
CONST      000017
OPCODE     201000
START      000000'
```

THE CREF PROGRAM

The following is an example of a listing you receive after running CREF with the preceding .CRF file:

```

^LCREFD CREF Example  MACRO %53A(1152) 13:58 10-Sep-79 Page 1
CREFD  MAC 10-Sep-79 13:44 Show Internal CREF Information

1  TITLE CREFD CREF Example
2  SUBTTL Show Internal CREF Information
3
4  SEARCH MACSYM, MONSYM ;Pseudo-op reference
5
6  CONST=17 000017 ;Symbol definition
7
8  DEFINE MACRO(X)< ;Pseudo-op reference, macro definition (in-line)
9  OPDEF X ;OPDEF and symbol reference (when expanded)
10
11
12 201000 000000 ;Pseudo-op reference, OPDEF definition and
13 ; opcode reference
14
15 START: MACRO CONST ;Symbol and macro reference
16 000000' 201 00 0 00 000017 ;OPDEF and symbol reference (when expanded)
17
18 000000' END START ;Pseudo-op and symbol reference

NO ERRORS DETECTED

PROGRAM BREAK IS 000001
CPU TIME USED 00:00.873

67P CORE USED
^LCREFD CREF Example  MACRO %53A(1152) 13:58 10-Sep-79 Page S-1
CREFD  MAC 10-Sep-79 13:44 SYMBOL TABLE

CONST 000017
OPCODE 201000 000000
START 000000'

^LCONST 6# 16
START 15# 18

^LMACRO 8# 15
OPCODE 12# 16

^LDEFINE 8
END 18
MOVEI 12
OPDEF 12
SEARCH 4
SUBTTL 2
TITLE 1

```

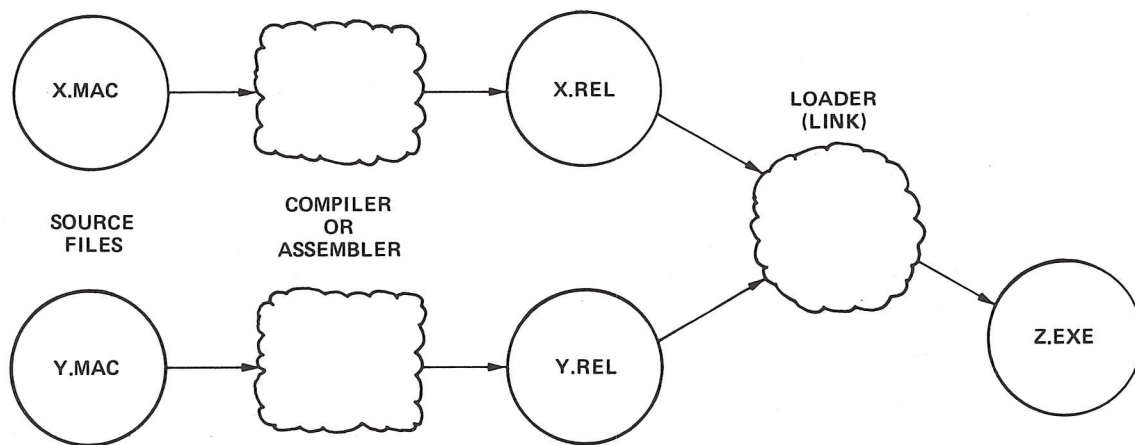

CHAPTER 6

THE MAKLIB PROGRAM

6.1 INTRODUCTION

The MAKLIB program organizes and manipulates files of relocatable object (REL) modules. These REL modules are the output from a source language translator, such as the FORTRAN compiler or the MACRO assembler.

At load time, the modules are linked together to build an executable program. As shown in Figure 6-1, the MACRO assembler processes two source files, X.MAC and Y.MAC, and produces two corresponding .REL files, X.REL and Y.REL. The LINK program then loads the resulting object modules from these .REL files and produces an executable program, Z.EXE.



MR-S-588-80

Figure 6-1 Generation of an .EXE File

Multiple modules can be concatenated into a single file called a library. A file containing a single module can also be called a library. (See Figure 6-2.)

THE MAKLIB PROGRAM

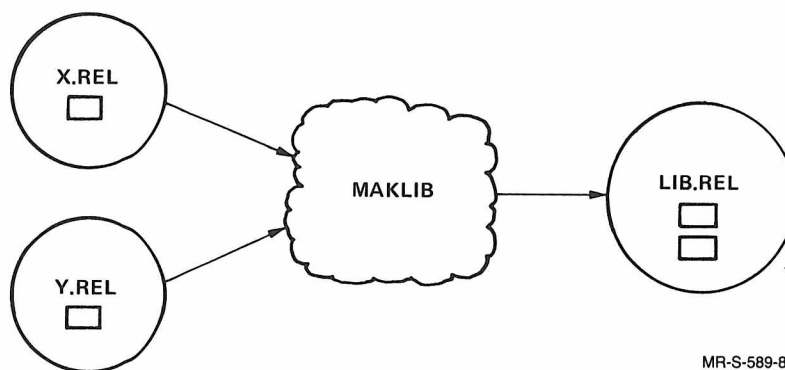


Figure 6-2 Generation of a Library

MAKLIB performs four functions on library files. Each function has switches that cause the MAKLIB program to do a specific task. The four functions are:

Obtaining Information about Libraries

These switches cause MAKLIB to give information about the status and contents of the library.

Manipulating Libraries

These switches cause MAKLIB to create new libraries by combining modules. Other switches cause MAKLIB to add, delete, extract, or replace modules.

Modifying Libraries

These switches cause MAKLIB to create new libraries from existing libraries, either by adding an index or removing local symbols. By modifying libraries you can reduce the amount of processing time required by the LINK program.

Editing Libraries

These switches cause MAKLIB to edit (or patch) modules within the library. You selectively change the object code in a module by supplying MAKLIB with the required MACRO assembly language code changes.

For more information on the contents of .REL files and REL Block types, refer to the TOPS-20 LINK Reference Manual. For more information on MACRO assembly language, refer to the TOPS-20 MACRO ASSEMBLER Reference Manual.

THE MAKLIB PROGRAM

6.2 RUNNING MAKLIB

To run MAKLIB, type MAKLIB after the TOPS-20 prompt (@) and press the RETURN key. The program prompts you with an asterisk:

```
@MAKLIB RET
*
```

After this prompt, enter a command string. MAKLIB takes commands in the following format:

```
Destination file spec=Source file spec1/Switches,Source file
spec2/Switches... Source file specn/Switches
```

where:

Destination file specification is the output file that MAKLIB produces. It can be either a text file or a library, depending upon the function you perform.

If you do not specify a Destination file name, MAKLIB uses the name of the file in Source file spec1. If you omit the Destination file type, the default depends on the function you perform.

Source file spec1 is the master library. This file specification is always required in a MAKLIB command string.

You must specify a file name in Source file spec1. The default file type is always .REL.

Source file spec2 ...Source file specn are the transaction files. These are additional input files required to perform some MAKLIB functions. A function usually requires only one transaction file.

You include switches in the command string to instruct MAKLIB to perform a specific function. You specify switches in one of the following formats:

```
/Switch
/Switch:argument
/Switch:(arg1,arg2...argn)
```

You can perform only one action with a switch in a single command string, but MAKLIB allows up to 100 switch arguments for each command string.

You can use MAKLIB switches in abbreviated form as long as they remain unique. However, arguments to switches are usually module names and hence cannot be abbreviated. Parentheses are optional when you specify only one argument, but are required to enclose two or more switch arguments.

Table 6-1 (in Section 3) is an alphabetical list of the MAKLIB switches; they are discussed in detail in Sections 6.2.1 through 6.2.4.

You can use an indirect file in a MAKLIB command string to reference another file. The indirect file can contain a complete or partial MAKLIB command string (file names and switches). For more information on using indirect files, refer to the TOPS-20 User's Guide.

If you always want to use specific switches when you run MAKLIB, you can put these switches in a SWITCH.INI file. For more information on creating a SWITCH.INI file, refer to the TOPS-20 User's Guide.

THE MAKLIB PROGRAM

MAKLIB allows you to type a string of commands on one or more lines. If the command string takes more than one line, type a hyphen (-) at the end of the first line, and press RETURN. Then, when MAKLIB prompts with a pound sign (#), continue to type the command string on the next line. You can also use multi-line commands in an indirect file.

To exit from MAKLIB and return to TOPS-20 command level, type either /EXIT or CTRL/Z after the asterisk prompt.

6.2.1 Running MAKLIB to Obtain Information About Libraries

MAKLIB contains four switches that allow you to obtain information on the status and contents of the master library (first input file). The four switches are: /LIST, /POINTS, /TRACE, and /LOAD.

Command String Requirements -

Files: A master library and an output file are required in the command string for each of the four switches. None of the command strings require a transaction file.

Default file type: Output file type - .LST
Master library type - .REL

Arguments: (Modules affected by the switch) - None

The output file (.LST) is a text file that can be written to any output device that supports text files, such as TTY: or LPT:. The discussion of /POINTS in this section illustrates the use of this option.

/LIST - LIST Switch

This switch lists the names of the modules that are contained in the master library. In addition to the names, MAKLIB also lists the two data values from the END block (REL Block type 5) of the module. If the module is a two-segment program, the first value is the high-segment break, and the second value is the low-segment break. If the module is a one-segment program, the first value is the program break, and the second value is the absolute break. If the second value is zero, it is not printed.

For example, if you want to create a file, REAP.LST, showing the names of all modules in the library IAGO.REL, give the following command string:

```
@MAKLIB RET
*REAP.LST=IAGO.REL/LIST RET
*
```

When MAKLIB finishes the task you request, it prompts you with another asterisk. You can now enter another command string.

THE MAKLIB PROGRAM

You get the following when you type the new file:

```
@TYPE (FILE) REAP.LST RET
Listing of Modules
Produced by MAKLIB Version 2A (67) on 12-Jun-79 at 14:02:41
```

```
DSK:IAGO.REL[4,244] created on 8-Jun-79 at 14:32:00
```

```
IAGO 401725 023746
@
```

The output file REAP.LST shows that the file IAGO.REL contains one module named IAGO. This module is a two-segment program. The first value listed, 401725, is the high-segment break. The second value listed, 023746, is the low-segment break.

/POINTS - POINTS Switch

This switch lists all entry points in the specified library. Entry points are usually subroutine starting addresses. They are used by the LINK program to determine if a global request can be satisfied by loading a module from a library.

For example, if you want to know all the entry points in the library NICE.REL and have the output written to the device TTY: (your terminal), do the following:

```
@MAKLIB RET
*TTY:=NICE.REL/POINTS RET
Listing of Modules and Entry Points
Produced by MAKLIB Version 2A (67) on 12-Jun-79 at 16:33:45
```

```
DSK:NICE.REL[4,244] created on 12-Jun-79 at 16:11:00
```

```
NICE BEGIN LOOP NICE
*
```

In this example, the output shows that the library NICE.REL contains one module, NICE, which has three entry points: BEGIN, LOOP, and NICE. After the output, MAKLIB returns with the asterisk prompt and waits for another command string.

/TRACE - TRACE Switch

This switch lists all the edits you have made to a library. This information is contained in the TRACE blocks in the specified library. MAKLIB creates these TRACE blocks (REL Block type 1060) when you use /FIX to edit a module in the library. The TRACE blocks contain information about the edits you insert and the changes you make to the library. For more information on TRACE blocks, refer to Section 6.5. /TRACE allows you to determine the exact binary patching status of the library.

THE MAKLIB PROGRAM

For example, if you want to see all the edits in the library OLDLIB.REL, and have the output written to the device TTY:, do the following:

```
@MAKLIB RET
```

```
*TTY:=OLDLIB.REL/TRACE RET
```

Listing of TRACE blocks

Produced by MAKLIB Version 2A(67) on 10-Jul-79 at 9:33:59

DSK:OLDLIB.REL[4,601] created on 1-Jun-79 at 9:31:00

Module: SORT Edit: 341

Status is Active

Last affected by DRB

Created by HAS On 2-Jun-79

Installed by DRB On 10-Jul-79

Program changes:

Inserts 4 instruction(s) at location 001046

This example shows that the module SORT in the library OLDLIB.REL has one edit inserted. The status of the edit is active. You can change the status of a particular edit with the .REMOVE or .REINSERT pseudo-ops. The example also shows that the edit was last affected by DRB. This information comes from /WHO when you install or change an edit. The edit was created by HAS on 2-Jun-79. This information comes from the .NAME and .DATE pseudo-ops, respectively. The edit was installed by DRB on 10-Jul-79. This information comes from /WHO and the date that the system supplies when you install an edit. For more information on these pseudo-ops and /WHO, refer to Section 6.2.4.

/LOAD - LOAD Switch

This switch shows additional loading instructions that are embedded within the library in either REQUEST (REL Block type 17), REQUIRE (REL Block type 16), or ASCII text blocks.

The library QUICK.REL was produced from a MACRO program containing the following statements:

```
TITLE      QUICK
.TEXT      "/VERSION:2(111)"
.REQUEST   SYS:FORLIB
.REQUIRE  SYS:MACREL
```

To see the additional loading instructions embedded within QUICK.REL in the blocks, do the following:

```
@MAKLIB RET
```

```
*TTY:=QUICK.REL/LOAD RET
```

Listing of Internal loading instructions

Produced by MAKLIB Version 2A(67) on 10-Jul-79 at 9:06:23

DSK:QUICK.REL[4,601] created on 6-Jul-79 at 13:28:00

Module: QUICK

Text strings: /VERSION:2(111)

Requests SYS:FORLIB

Requires SYS:MACREL

THE MAKLIB PROGRAM

6.2.2 Running MAKLIB to Manipulate Libraries

For handling and creating libraries, MAKLIB includes six switches that allow you to work with individual modules within libraries. The six switches are: /MASTER, /APPEND, /DELETE, /EXTRACT, /INSERT, and /REPLACE.

Command String Requirements -

Files: A master library (first input file) and an output file are required in the command string for each of the six switches. A transaction file is required with some switches.

Default file type: .REL for all files

Arguments: All switches accept arguments. /APPEND and /INSERT are two switches that do not always require arguments. For more information, refer to the discussions of these two switches in this section.

/MASTER - MASTER Switch

This switch identifies modules within the master library that correspond to those in the transaction file being used to effect the update. /MASTER takes at least one argument, and requires that another switch be given in the same command string. It is the only switch within this function that causes no real manipulation of a library. It is mentioned here because some of the switches used to manipulate libraries require /MASTER in their respective command strings.

You include /MASTER in the command strings for only two switches, /INSERT and /REPLACE. These switches are discussed later in this section.

/APPEND - APPEND Switch

This switch adds new modules to the end of an existing library. The output file is the master library plus the appended modules. MAKLIB reads these appended modules from the transaction file. You specify them as arguments to the switch. You must specify modules as arguments in the same physical order as they occur in the transaction file. Figure 6-3 illustrates the function of /APPEND. Note that, in the figure, modules D and E from the transaction file are appended to the master library.

THE MAKLIB PROGRAM

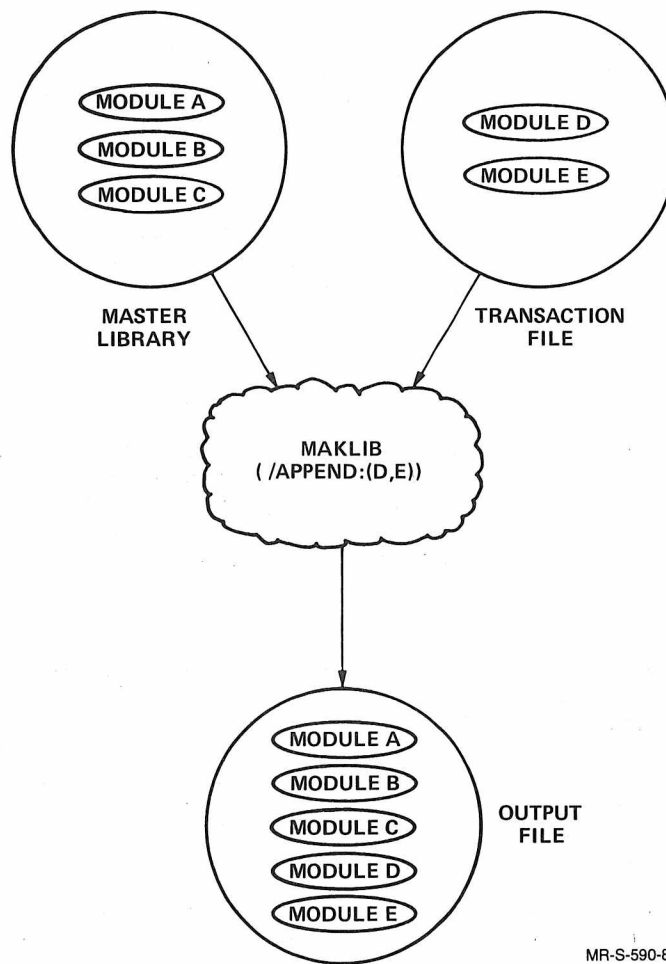


Figure 6-3 Function of /APPEND

NOTE

When you do not specify an argument to /APPEND, the entire transaction file is appended to the master library.

THE MAKLIB PROGRAM

For example, you want to append the module IAGO in the library IAGO.REL to the library GRAF.REL. You name the output file EXEN.REL. The command string is:

```
@MAKLIB RET
*EXEN.REL=GRAF.REL,IAGO.REL/APPEND:IAGO RET
*
```

MAKLIB returns with the asterisk prompt for you to enter another command string. To check the new file, use /LIST and have the output written to the device TTY:.

```
*TTY:=EXEN.REL/LIST RET
      Listing of Modules
Produced by MAKLIB Version 2A(67) on 14-Jun-79 at 10:45:13
```

DSK:EXEN.REL[4,244] created on 14-Jun-79 at 10:43:00

```
CRIG      004582
PASZ      003217
TENP      036651
IAGO      402420    023746
*
```

The example shows that the new file, EXEN.REL, contains four modules: the three modules from the library GRAF.REL (CRIG, PASZ, TENP), and the appended module IAGO.

/APPEND also allows you to initially create a library. For example, you wish to combine the following four .REL files into a single library:

```
GLOBAL.REL
DTABLE.REL
INOUT.REL
IO.REL
```

Type the following MAKLIB command string to produce the file LIB.REL, which contains all modules from the four specified files:

```
@MAKLIB RET
*LIB=GLOBAL,DTABLE/APPEND,INOUT/APPEND,IO/APPEND RET
*
```

In this example, MAKLIB copies the file GLOBAL to a file named LIB, and then appends all modules from the files DTABLE, INOUT, and IO to LIB. This also illustrates the use of more than one transaction file.

/DELETE - DELETE Switch

This switch removes one or more modules from an existing library. The output file is the master library minus the deleted module(s). All modules, except those specified as arguments to /DELETE, are read from the master library and copied to the output file. No transaction file is required.

THE MAKLIB PROGRAM

NOTE

You must specify modules as arguments in the same physical order as they occur in the master library.

Figure 6-4 illustrates the function of /DELETE. Note that, in the figure, module B is deleted from the master library.

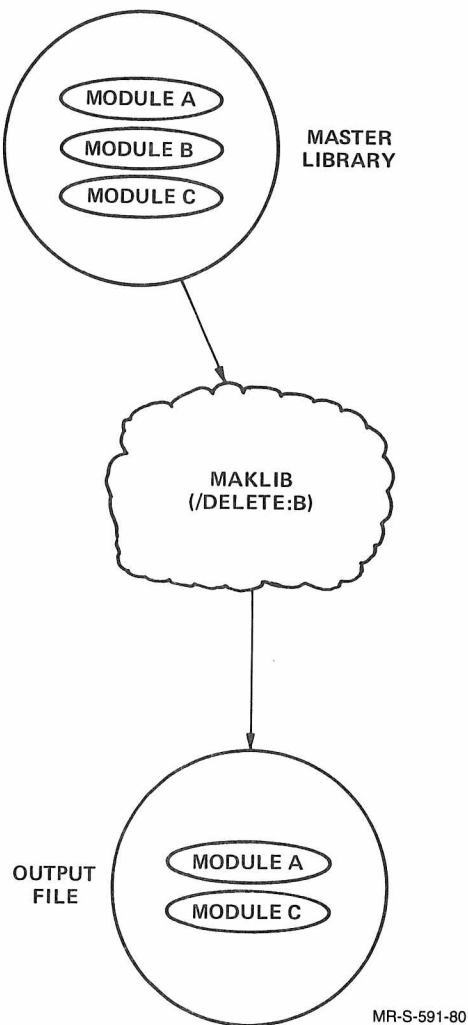


Figure 6-4 Function of /DELETE

THE MAKLIB PROGRAM

Type the following command string to delete the module TENP from the library EXEN.REL. The output filename is DIPEP.REL.

```
@MAKLIB RET
*DIPEP.REL=EXEN.REL/DELETE:(TENP) RET
*
```

The program returns with the asterisk prompt for you to enter another command string. To check the new file, use /LIST and have the output written to the device TTY:.

```
*TTY:=DIPEP.REL/LIST RET
Listing of Modules
Produced by MAKLIB Version 2A(67) on 14-Jun-79 at 14:10:41
```

DSK:DIPEP.REL[4,244] created on 14-Jun-79 at 14:10:00

```
CRIG      004582
PASZ      003217
IAGO      402420  023746
*
```

/EXTRACT - EXTRACT Switch

This switch produces an output file that is a subset of modules in the master library. You specify the modules as arguments to the switch. No transaction file is required.

NOTE

You must specify the modules as arguments in the same physical order as they occur in the master library.

Figure 6-5 illustrates the function of /EXTRACT. Note that, in the figure, module A is extracted from the master library.

THE MAKLIB PROGRAM

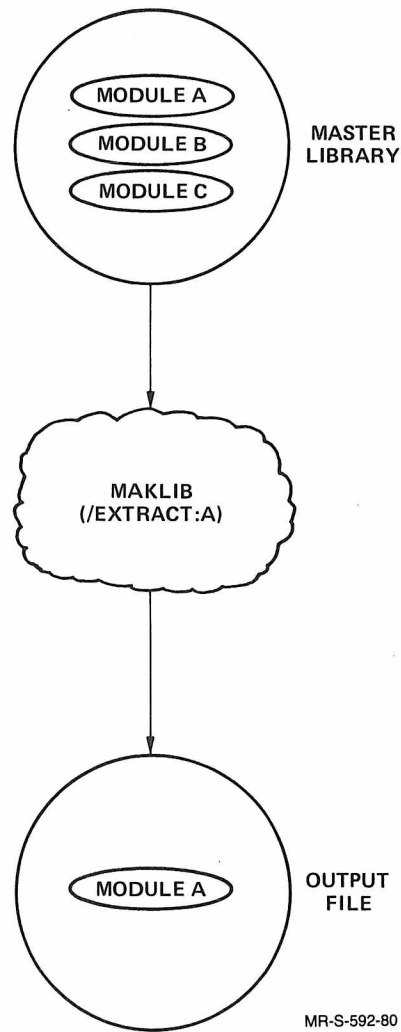


Figure 6-5 Function of /EXTRACT

THE MAKLIB PROGRAM

The following example illustrates /EXTRACT. The library FOO.REL contains four modules. To get a listing of the module names, use /LIST and have the output written to the device TTY:.

```
@MAKLIB RET
*TTY:=FOO.REL/LIST RET
      Listing of Modules
Produced by MAKLIB Version 2A(67) on 18-Jun-79 at 14:37:45

*****

DSK:FOO.REL[4,244] created on 18-Jun-79 at 13:54:00

SQUARE 000023
BOX     000014
MAIN    000433
DRAW    000505
*
```

You want to extract two of these modules, SQUARE and BOX, from the library FOO.REL, and put them in a new library, 2FOO.REL. After using /EXTRACT, check on the new file with /LIST, and have the output written to the device TTY:. Type the MAKLIB command string after the asterisk prompt:

```
*2FOO.REL=FOO.REL/EXTRACT:(SQUARE,BOX) RET
*TTY:=2FOO.REL/LIST RET
      Listing of Modules
Produced by MAKLIB Version 2A(67) on 18-Jun-79 at 14:57:17

*****

DSK:2FOO.REL[4,244] created on 18-Jun-79 at 14:56:00

SQUARE 000023
BOX     000014
*
```

The example shows that the new library, 2FOO.REL, contains the two modules that you extracted from the master library, FOO.REL.

/INSERT - INSERT Switch

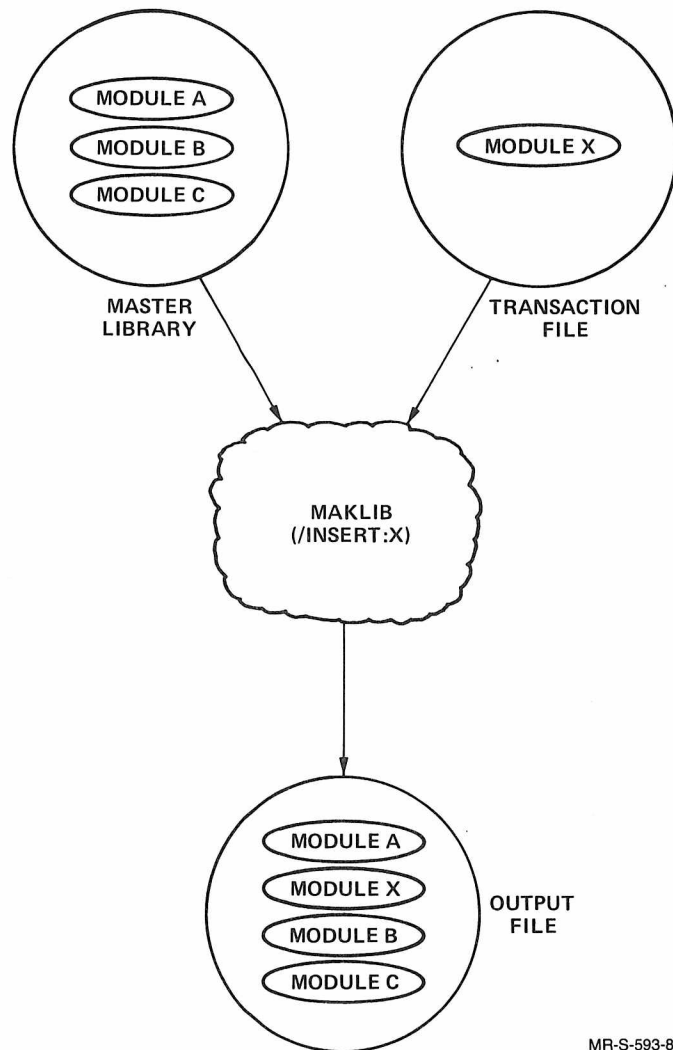
This switch inserts new modules into a master library. /MASTER is required in the command string. The output file is formed as follows: MAKLIB copies the master library to the output file up to but not including the module named as the first argument to /MASTER. Next, MAKLIB copies the module named as the first argument to /INSERT from the transaction file to the output file. The process repeats until the argument list specified to /MASTER and /INSERT is exhausted. At this point, MAKLIB copies the remaining modules in the master library to the output file. There must be one argument to /MASTER for every argument to /INSERT.

THE MAKLIB PROGRAM

NOTE

You must specify the module names in the argument lists for /MASTER and /INSERT in the same physical order as they occur in the master library and the transaction file, respectively. When you do not specify an argument to /INSERT, the entire transaction file is inserted before the module you specify to /MASTER. You must always specify an argument to /MASTER.

Figure 6-6 illustrates this function of /INSERT. Note that, in the figure, module X in the transaction file is inserted before module B in the master library.



MR-S-593-80

Figure 6-6 One Function of /INSERT

THE MAKLIB PROGRAM

For example, the library FOO.REL contains four modules: SQUARE, BOX, MAIN, and DRAW. The library NICE.REL contains one module, NICE. You want to insert the module NICE before the module BOX in FOO.REL. The name of the output file containing the five modules is CLAR.REL. The command string to insert this module is as follows:

```
@MAKLIB RET
*CLAR.REL=FOO.REL/MASTER:BOX,NICE.REL/INSERT:NICE RET
*
```

MAKLIB returns with the asterisk prompt. You can now check on the new library, CLAR.REL, with /LIST. The output from /LIST is written to the device TTY:

```
*TTY:=CLAR.REL/LIST RET
Listing of Modules
Produced by MAKLIB Version 2A(67) on 27-Jun-79 at 15:40:28
```

DSK:CLAR.REL[4,244] created on 27-Jun-79 at 15:40:00

```
SQUARE 000023
NICE    005557
BOX     000014
MAIN    000433
DRAW    000505
*
```

You may also insert more than one module in front of a module in a master library. However, the master library module name must appear repeatedly in the argument list to /MASTER. This produces a one-to-one correspondence between the module in the master library and the modules you wish to insert. In this case, you must list the argument names for both /MASTER and /INSERT in the same physical order that they appear as modules in the master library and transaction file, respectively.

Figure 6-7 illustrates this function of /INSERT. Note that, in the figure, modules X and Y in the transaction file are inserted before module B in the master library.

THE MAKLIB PROGRAM

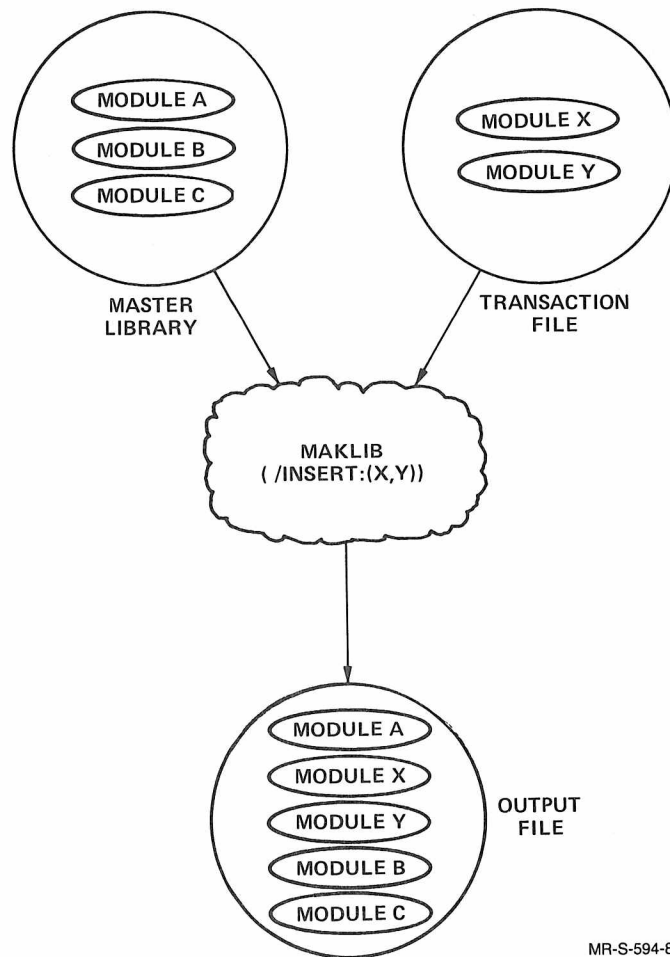


Figure 6-7 One Function of `/INSERT`

THE MAKLIB PROGRAM

For example, the library SFJA.REL contains two modules: ILJA, and HLBET. The library FOO.REL contains four modules: SQUARE, BOX, MAIN, DRAW. You want to insert both modules in SFJA.REL in the library FOO.REL, before the module DRAW. The name of the new library is SFOO.REL. The command string is:

```
@MAKLIB RET
*SFOO.REL=FOO.REL/MASTER:(DRAW,DRAW),SFJA.REL/INSERT:(ILJA,HLBET) RET
*
```

After the asterisk prompt, check on the contents of the new library, SFOO.REL, with /LIST and have the output written to the device TTY:.

```
*TTY:=SFOO.REL/LIST RET
Listing of Modules
Produced by MAKLIB Version 2A(67) on 28-Jun-79 at 16:30:22
```

DSK:SFOO.REL[4,244] created on 28-Jun-79 at 16:30:00

```
SQUARE 000023
NICE    005557
BOX      000014
MAIN     000433
ILJA     001047
HLBET    000433
DRAW     000505
*
```

/REPLACE - REPLACE Switch

This switch replaces modules in the master library with those specified in the transaction file. /MASTER is required in the command string so that the program can identify the modules in the master library that are to be replaced by those named as arguments to /REPLACE. There must be a one-to-one correspondence between the number of arguments to /MASTER and /REPLACE.

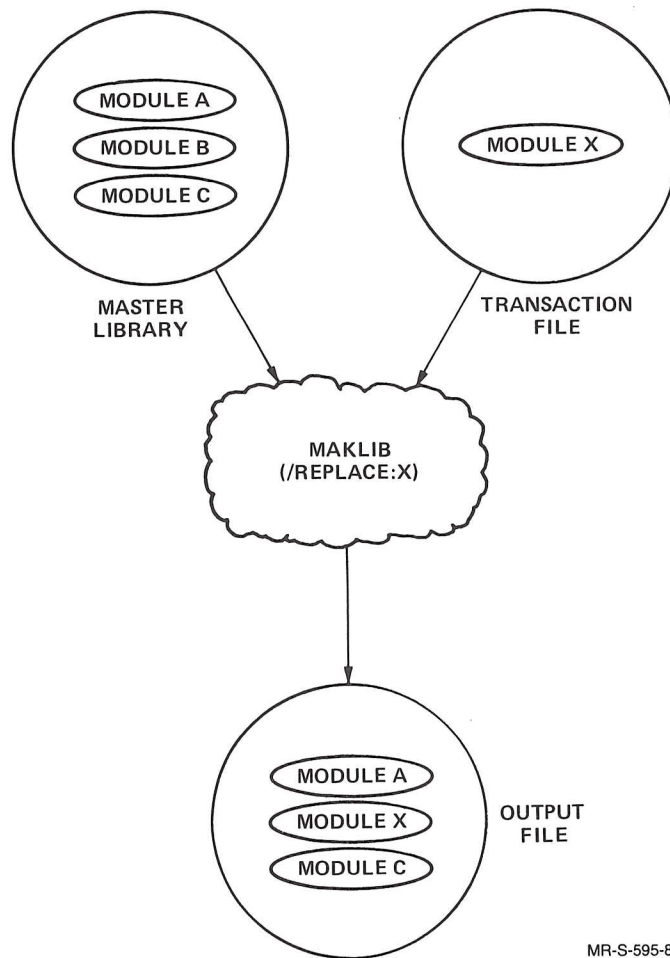
The output file is the entire master library, with modules replaced by those read from the transaction file (and named as arguments to the switch).

NOTE

You must specify the names in both argument lists (/MASTER and /REPLACE) in the same physical order as they appear as modules in the master library and transaction file, respectively.

Figure 6-8 illustrates the function of /REPLACE. Note that module X in the transaction file replaces module B in the master library.

THE MAKLIB PROGRAM



MR-S-595-80

Figure 6-8 Function of /REPLACE

For example, the library FOO.REL contains four modules: SQUARE, BOX, MAIN, and DRAW. The library NICE.REL contains one module, NICE. You want to replace the module MAIN in FOO.REL with the module NICE in NICE.REL. The output file name is WELW.REL. The command string is the following:

```
@MAKLIB RET
*WELW.REL=FOO.REL/MASTER:MAIN,NICE.REL/REPLACE:NICE RET
*
```


THE MAKLIB PROGRAM

After MAKLIB completes the action you requested, it prompts you with an asterisk. Now check on the contents of the new library, WELW.REL, with /LIST and have the output written to the device TTY:.

```
*TTY:=WELW.REL/LISTRET
```

Listing of Modules

Produced by MAKLIB Version 2A(67) on 6-Jul-79 at 14:49:22

DSK:WELW.REL[4,244] created on 6-Jul-79 at 14:48:00

SQUARE 000023

BOX 000014

NICE 005557

DRAW 000505

*

NICE replaced MAIN in the master library FOO.REL. The new library, WELW.REL, contains the four modules: SQUARE, BOX, NICE, and DRAW.

6.2.3 Running MAKLIB to Modify Libraries

The two switches within this function facilitate the processing of requests by the LINK program when it loads modules from libraries. The two switches are: /INDEX and /NOLOCALS.

Command String Requirements -

Files: A master library (first input file) and an output file are required in the command string for both switches. Transaction files are not allowed. Both switches appear with the master library in the command string.

Default file type: .REL for both the master library and the output file for both switch command strings.

Arguments: None

/INDEX - INDEX Switch

This switch produces an output file, which is identical to the master library, except with INDEX blocks (REL Block type 14) inserted in the file. Normally, programs make external requests to library subroutines they need, and LINK must search completely through the library to decide which modules to load to satisfy the requests. INDEX blocks list the entry point names and corresponding modules, allowing LINK to quickly determine which modules to load. LINK searches more efficiently, and loading time is shorter because the amount of I/O is reduced.

THE MAKLIB PROGRAM

/NOLOCALS - NOLOCALS Switch

This switch produces an output file which is the master library with all local symbols deleted from the file SYMBOL blocks (REL Block type 2). Local symbols are useful for debugging purposes, and also when modules are edited with MAKLIB. (Refer to FIX files, Section 6.2.4.) In a production-mode library, local symbols are usually deleted, because they serve no purpose. This reduces the amount of mass storage space the library occupies. In addition, loading time is faster because the amount of I/O is reduced. Global symbols are not deleted because they are used in the linking of modules.

In the following example you create a new library, L2, from L1. The new library has an index but no local symbols. You can use /INDEX and /NOLOCALS together in the command string.

```
@MAKLIB RET
*L2=L1/NOLOCALS/INDEX RET
*
```

6.2.4 Running MAKLIB to Edit Libraries

MAKLIB provides a mechanism for you to patch (or edit) the code of a relocatable object module. This patching facility allows you to make program changes directly to a library. Although MAKLIB provides the facility for editing a program without having to change the source code, good programming practice requires that the same edits also be made at the source level.

To edit a library in this way, you must first create a text file called a .FIX file. This .FIX file contains one or more edits that you want to insert in the library. Each edit has a unique identifier and consists of a sequence of control pseudo-ops and MACRO assembly language code. The control pseudo-ops tell MAKLIB where and how to make the changes. Any new code is supplied as a sequence of MACRO assembly language statements. Each edit begins with a .EDIT pseudo-op and ends with a .ENDE pseudo-op. Figure 6-9 shows the order that pseudo-ops must appear within an edit.

When MAKLIB processes the .FIX file, it creates a new library with any new edits inserted. Although each edit is now a permanent part of the library, you can use MAKLIB to deactivate an edit. This operation effectively removes any code changes inserted by the edit. Therefore, edits in the library can be either active or inactive.

MAKLIB maintains edit history information on the library by generating TRACE blocks (REL Block type 1060) for each edit you insert. TRACE blocks are part of the module. Therefore, when you use /REPLACE, /EXTRACT, or /DELETE, the TRACE blocks move with the module. Section 6.2.1 describes how you can determine the edit status of the library using /TRACE.

NOTE

MAKLIB does not handle PSECTS.

THE MAKLIB PROGRAM

Pseudo-ops for .FIX files

.EDIT xxxxxx - This pseudo-op is an identifying name for the edit you insert in the specified module. The edit name (xxxxxx) can be up to six (6) SIXBIT characters and is stored in the TRACE block for any module affected by the edit. .EDIT is the first pseudo-op for each edit.

.DATE dd-mmm-yy - This pseudo-op gives the date that the edit was made. The day (dd) and year (yy) entries are numeric. The month entry (mmm) is alphabetic. .DATE is an optional pseudo-op. If you supply this information, it is stored in the edit TRACE block.

.NAME xxx - This pseudo-op gives the three initials (xxx) of the person who wrote the edit. .NAME is an optional pseudo-op. If you supply these initials, they are stored in the TRACE block for the edit.

.MODULE xxxxxx - This pseudo-op gives the name (xxxxxx) of the module that you wish to edit. It is the module name as it appears in the library, up to six Radix-50 characters. Once you give a name, that module is loaded. Editing continues with this module unless a new .MODULE pseudo-op is given. You do not have to edit modules in the same order that they reside in the master library (first input file). However, each module may be named only once within an edit.

.ASSOCIATED +edit1,-edit2,+edit3,+edit4..... - This pseudo-op gives information on which other related edits must be present in the specified module. The edit names here are the same as those designated under the .EDIT pseudo-op. The "+" indicates that the edit is required. The "-" indicates an edit that conflicts with the current edit; you cannot have both edits active simultaneously. You receive notification if the specified module does not contain the correct combination of associated edits. The default is "+" if no sign precedes the edit name. Information you supply with the .ASSOCIATED pseudo-op must precede any information supplied in the .FIX file by the .INSERT, .REMOVE, or .REINSERT pseudo-ops.

.VERSION nnnxx(nnnnnn)-g - This pseudo-op allows you to specify the version number of the edit. The version is in standard TOPS-20 version format. The nnn designates the major version number, which consists of three octal digits maximum. The xx designates the minor version, which consists of two letters maximum. The (nnnnnn) designates the edit number, and can be up to six octal digits. The g designates the code for the group that last edited the module, and it is one octal digit maximum. All fields are optional.

.ALTER location,<new value>,<original value> - This pseudo-op changes the contents of a specific location. All code is written in angle brackets, < >. The original value at the specified location is replaced by a new value. The first argument, location, is where you wish to place the new value. Once you enter the new value, it is evaluated and placed into the specified location. You can specify a third argument, <original value>, to check whether the actual original value differs from the expected original value. If it does, MAKLIB gives you an error message and the location is not altered.

THE MAKLIB PROGRAM

.INSERT location,keyword:n,<original value> - This pseudo-op allows you to add code to a module. You precede the new code with a .INSERT pseudo-op and terminate the sequence with a .ENDI pseudo-op. MAKLIB assembles the code and adds it to the module in the output file.

The .INSERT pseudo-op takes three arguments. The first gives the location at which you want the new code executed. This argument can be a numeric or symbolic expression. This location is assumed to be relocatable, and may be in either the low or the high segment. The second argument is a keyword that specifies how the code is to be located with respect to this location. This keyword is one of the following:

- BEFORE - Insert the new instructions so that they are executed before the instruction at the specified location.
- AFTER - Insert the new instructions so that they are executed after the instruction at the specified location.
- REPLACE - Delete one instruction from the existing code for each one included in the edit, beginning with the instruction at the location of the edit. Then, insert the new instructions so that they are executed in place of the deleted code.
- REPLACE:n - Delete n instructions from the existing code, starting with the instruction at the edit address. Then, insert the new instructions so that they are executed in place of the deleted code. This applies no matter how many instructions you insert. The argument may be an expression, and is evaluated in the current radix.

You can specify a third argument, <original value>, to check whether an actual original value differs from the expected original value. If it does, MAKLIB gives you an error message and the editing does not take place. This argument gives the line of code at the location specified by the first argument. The code is written in angle brackets, and is evaluated. It must exactly match the code at the specified location. If the code at the location is a literal, you must give only the first word.

MAKLIB always inserts the new code at the end of the current segment. It replaces the referenced instruction with an unconditional jump to the new code. The format of code insertion appears in Section 6.5, Technical Notes.

Because MAKLIB does not physically insert the code at the location you specify, you need to consider a restriction when you use this facility.

MAKLIB constructs a patch that has the effect of a skipping instruction, and assumes that it follows an instruction that can potentially skip, at most, one instruction. If the intent of the patch does not fit these assumptions, it may not work correctly. Further, for REPLACE functions, MAKLIB assumes that program control can only pass to the first instruction of the deleted code.

THE MAKLIB PROGRAM

For example consider the following code segment,

```
        JRST    FOO
BAR:    JFCL
        JFCL
FOO:    JFCL
        JFCL
        JFCL
```

and a MAKLIB patch to it:

```
.INSERT BAR,REPLACE:4<JFCL>
      JFCL
.ENDI
```

Note that the JRST FOO instruction at BAR-1 still causes the old code to be executed in spite of the patch.

As another example, consider the following:

```
LABEL:  OPENF%      ;A JSYS MONITOR CALL
        ERJMP ERROR
        JFCL
```

A patch using .INSERT cannot specify any combination of BEFORE or AFTER with location LABEL or LABEL+1. To do so would separate the JSYS and the ERJMP instructions, which must be consecutive to operate properly.

.REMOVE edit1,edit2,edit3.... - This pseudo-op deactivates the specified edits from the selected module. The original instructions displaced by the jumps to the edit area for each .INSERT are returned to that location. No changes are made to the symbol table. The arguments are the edits you wish to remove.

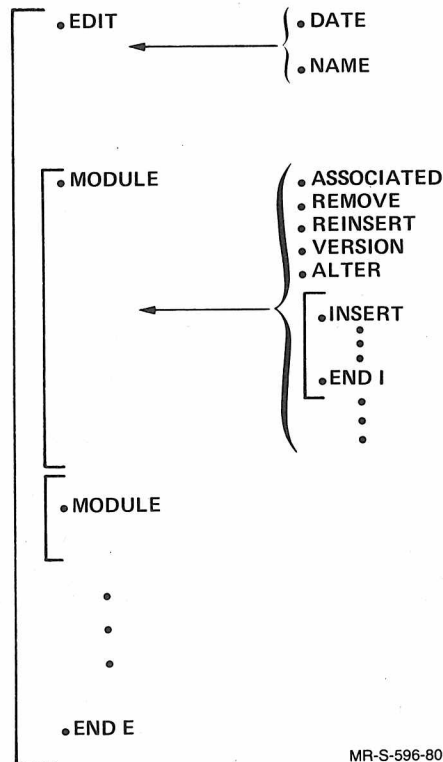
.REINSERT edit1,edit2,edit3... - This pseudo-op activates any edits you previously removed with the .REMOVE pseudo-op.

.ENDI - This pseudo-op marks the ending point of the code following the last .INSERT pseudo-op.

.ENDE - This pseudo-op marks the ending point of the complete edit. It also instructs MAKLIB to check for undefined labels or other invalid entries within the edit.

THE MAKLIB PROGRAM

Figure 6-9 illustrates the order that pseudo-ops appear in a .FIX file:



MR-S-596-80

Figure 6-9 Order of Pseudo-ops in a .FIX File

The MAKLIB program has a one-pass assembler. Because of this, forward references to labels and expressions are restricted to simple addition and subtraction on the halfword boundary. References to undefined labels or symbols are valid where references to external symbols would be valid in MACRO (with no polish fix-ups). Literals are treated as forward references, because the actual location of the literal is not known until the .INSERT pseudo-op ends. Defining a label inside of a literal is not valid. Finally, the value you place in the right-hand side of an assignment must not be forward or external.

It is not required that assignments be inside .INSERT in the .FIX file. It is required, however, that the .EDIT and .MODULE pseudo-ops precede any assignments, because these define new symbols in the symbol table. MAKLIB does not allow redefinitions of existing symbols, because it is impossible to backtrack references to a symbol in the relocatable binary file. So, any label or symbol you create with /FIX must be new to the program.

THE MAKLIB PROGRAM

To simplify editing and to keep the appearance of binary edits as close as possible to the source level, the following pseudo-ops are implemented in the MAKLIB .FIX file assembler and operate as they do in MACRO:

| | | |
|---------|---------|--------|
| ASCII | ASCIZ | BLOCK |
| BYTE | COMMENT | DEC |
| EXP | IOWD | OCT |
| POINT | PURGE | RADIX |
| RADIX50 | REMARK | SIXBIT |
| SQUOZE | SUBTTL | TITLE |
| XWD | | |

NOTE

The pseudo-ops BYTE, DEC, OCT, and EXP are limited to a maximum generation of one word of data.

All MACRO operators and qualifiers are available except ^F. MAKLIB also supports the following MACRO pseudo-ops for writing conditional code:

| | | |
|-----|------|--------|
| IFN | IFG | IFDEF |
| IFE | IFLE | IFNDEF |
| IFL | IFGE | |

You may follow symbols with ## (double pound sign) to indicate that they are EXTERNAL quantities. However, if the symbol is defined as EXTERNAL (already in the symbol table), you do not have to use ##. It is not necessary to follow undefined symbol names with # (single pound sign), since it is assumed that any undefined symbol is a forward reference. If a symbol name is already assigned and followed by the #, you receive an error message (see Section 6.4, MAKLIB Messages). You may define labels as internal (available to other programs) if they are followed by :: (double colon). Entry points may not be defined. The full facilities available in MACRO for combinations of DDT suppression and internal declaration are available for both labels and assignments.

Command String Requirements -

Files: A master library (first input file) and an output file are required in the command string. The .FIX file is the required transaction file.

Default file type: .REL for both the output file and the master library. The default for the transaction file is .FIX.

Arguments: None.

The editing command string accepts two switches. They are /FIX and /WHO.

/FIX - FIX Switch

This switch makes changes to the actual code and symbol table of a module. It appears with the transaction (.FIX) file specification.

THE MAKLIB PROGRAM

/WHO:xxx - WHO Switch

This switch is optional and you use it only with /FIX. You can enter it in either the master library or the transaction (.FIX) file specification. The argument to /WHO can be up to three characters (xxx). These are usually the initials of the person using MAKLIB at the time the edit is installed. If you include this information, it appears in the TRACE block of all new edits (in the last affected field and in the last installed field). If any of these edits change the status of an existing edit (such as .REMOVE or .REINSERT), this information is entered in the last affected field of the TRACE block of the affected edit. If you use /WHO without /FIX, MAKLIB ignores it in the command string.

In order to edit a library with a .FIX file, you can use the following command string. In this example, you use the .FIX file FIX1.FIX to edit the library OLDLIB.REL, and create an updated library NEWLIB.REL.

```
@MAKLIB RET  
*NEWLIB=OLDLIB, FIX1/FIX/WHO:SFA RET  
*
```

The following are sample .FIX files. This first example illustrates the use of the .INSERT and .VERSION pseudo-ops. One module in the library is modified.

```
.EDIT 341  
.NAME HAS  
.MODULE GLOB..  
.VERSION 4C(341)  
.INSERT START,AFTER,<RESET>  
MOVE      P,[IOWD PDLEN,PDLIST]  
.ENDI  
.INSERT    LOOP+3,BEFORE,<JRST LOOP>  
          PUSHJ      P,NEXTCR  
.ENDI  
.ENDE
```

The following edit illustrates the use of .ALTER pseudo-op to change the value of a table entry. Location RAD50+46 is changed from a "." to a "\$". In addition, this edit uses the .ASSOCIATED pseudo-op to specify that this edit also requires edit 343 to the module TABLES.

```
.EDIT 344  
.NAME HAS  
.MODULE TABLES  
.ASSOCIATED 343  
.ALTER RAD50+46,<"$">,<". ">  
.ENDE
```

This edit uses the .REMOVE pseudo-op to deactivate edit 345 in the module FSORT. As a result of this operation, any code that was changed by edit 345 will be restored to its previous state.

```
.EDIT 346  
.NAME HAS  
.MODULE FSORT  
.REMOVE 345  
.ENDE
```

THE MAKLIB PROGRAM

6.3 MAKLIB SWITCH OPTIONS

Table 6-1 is a reference table of all MAKLIB program switches. These switches are listed alphabetically. The function that each switch performs appears beside it in the table.

Table 6-1
MAKLIB Switches

| Switch | Function |
|-----------|--|
| /APPEND | Adds new modules to the end of an existing library. |
| /DELETE | Removes one or more modules from an existing library. |
| /EXIT | Terminates MAKLIB and returns you to TOPS-20 command level. |
| /EXTRACT | Produces an output file that is a subset of modules in the master library. |
| /FIX | Makes changes to the actual code and symbol table of a module. |
| /INDEX | Produces an output file identical to the master library except with INDEX blocks inserted in the file. |
| /INSERT | Inserts new modules into the master library. |
| /LIST | Lists the names of the modules that are contained in the master library. |
| /LOAD | Shows additional loading instructions that are embedded within the library in either REQUEST, REQUIRE, or ASCII text blocks. |
| /MASTER | Identifies files within the master library that correspond to those in the transaction file being used to effect the update. |
| /NOLOCALS | Produces an output file which is the master library with all local symbols deleted from the file symbol blocks. |
| /POINTS | Lists all entry points in the specified library. |
| /REPLACE | Replaces modules in the master library with those specified in the transaction file. |
| /TRACE | Lists all the edits made to a library. |
| /WHO | Specifies the initials of the person using MAKLIB when an edit is installed. |

THE MAKLIB PROGRAM

6.4 MAKLIB MESSAGES

The MAKLIB program issues two types of messages: fatal errors and warning messages. Fatal errors are preceded by a question mark (?) and cause the current command to be aborted. Warning messages are preceded by a percent sign (%) and indicate that the command will be completed, but the operation may not have been performed as you intended.

All messages are typed on your terminal. They begin with a six character code that identifies the error. This is followed by a short description of the problem.

MAKLIB uses the command scanner routines SCAN and WILD. The following list of messages contains the most common messages that these routines produce. These messages begin with SCN or WLD.

Some of the messages contain information that is dependent on the exact command string, switch, or file you wish to process. The key to these message variables follows:

| | |
|-------------|---|
| [edit] | The name of a specific edit. |
| [file] | A file specification. |
| [label] | The name of the label which caused the error. |
| [location] | The location where the error was detected. This is expressed as either a symbolic address or as a line number in the .FIX file. |
| [module] | The name of a specific module. |
| [pseudo-op] | A specific pseudo-op. |
| [statement] | A specific statement related to or causing the error. |
| [status] | A specific numeric file status code. |
| [switch] | A specific MAKLIB command switch. |
| [symbol] | A symbol. (Refer to the <u>TOPS-20 MACRO ASSEMBLER Reference Manual</u> for an exact definition.) |
| [type] | A REL Block type. |
| [value] | A specific value. |

All MAKLIB messages are listed here alphabetically by the six-character code. A suggested user response is provided for each fatal error and those warning messages that require correction.

?MKLAAC .ASSOCIATED seen after .INSERT,.REMOVE or .REINSERT in edit
[edit]

Description: In the indicated .EDIT, the .ASSOCIATED pseudo-op is out of order.

Suggested User Response: Move the .ASSOCIATED pseudo-op so that it appears before any .INSERTs, .REMOVEs, or .REINSERTs.

THE MAKLIB PROGRAM

?MKLAAL .ALTER address is not in current module in edit [edit]

Description: In your edit, you used a .ALTER pseudo-op to change the contents of an address. However, you gave an address that does not exist in the specified module.

Suggested User Response: Change the .ALTER pseudo-op so that it specifies a legal address.

%MKLAFI Arguments to /FIX switch are ignored

Description: In the command string, you gave arguments on /FIX. There are no defined arguments for /FIX, so MAKLIB ignores any that you specify.

%MKLAMI Assignment to [symbol] with no module selected was ignored:
[statement]

Description: In a .FIX file you assigned a value to a symbol, but you did not specify a module.

Suggested User Response: Change the .FIX file so that the assignment statement occurs after the .MODULE pseudo-op. This specifies which symbol belongs with a particular module.

?MKLANA Asterisk not allowed as output file spec

Description: In the command string you gave an output file name that included a wildcard character.

Suggested User Response: Since wildcard characters are illegal for the output file name, reissue the command with an explicit output file name.

?MKLASG FORWARD/EXTERNAL assignment to [symbol] at [location] (Edit [edit])
[statement]

Description: In the specified edit, you assigned a forward or external reference.

Suggested User Response: MAKLIB does not support forward or external references. Change the assignment statement.

?MKLBAM BEFORE, AFTER or REPLACE missing from .INSERT in edit [edit]

Description: You typed an incomplete .INSERT pseudo-op in a .FIX file.

Suggested User Response: Include the required second argument for .INSERT indicating how the code should be inserted.

?MKLBDA Bad .DATE argument for edit [edit]

Description: In a .FIX file you specified a date that was not in a recognizable format for the .DATE pseudo-op.

Suggested User Response: Specify the date in the form dd-mon-yy, such as 7-JUL-77.

THE MAKLIB PROGRAM

?MKLBNI Binary patching tool not included in MAKLIB

Description: You attempted to use a .FIX file with a version of MAKLIB that does not support .FIX files.

Suggested User Response: Rebuild MAKLIB from the MACRO source files and set feature switch FTBPT non-zero.

?MKLCDM Existing code does not match original code

Description: The original value you specified with either the .ALTER or .INSERT pseudo-op does not match the actual code at the location you were attempting to change.

Suggested User Response: First, determine if this is an error in the original value field of the pseudo-op. If this is actually the value you expected at that location, this error could indicate that you are trying to edit a different version of the library file.

%MKLCII Code generated outside of range of .INSERT was ignored:
[statement]

Description: You entered a .FIX file with a sequence of code that was not included between .INSERT and .ENDI pseudo-ops.

Suggested User Response: Change the .FIX file so that the instructions are within the range of an .INSERT. They can then be inserted in the module.

%MKLCNF Insertion of edit [edit1] by edit [edit2] conflicts with edit [edit3]

Description: Edit2 contains a .REINSERT pseudo-op for edit1. This conflicts with the .ASSOCIATED list in edit3 which is currently an active edit for this module. This is only a warning message, and the actual .REINSERT does take place. You can verify the current status of any edits with /TRACE.

%MKLCNF Removal of edit [edit1] by edit [edit2] conflicts with edit [edit3]

Description: Edit2 contains a .REMOVE pseudo-op for edit1. This conflicts with the .ASSOCIATED list of edit3 which is currently an active edit for this module. This is only a warning message, and the actual .REMOVE does take place. To verify the status of any edits, use /TRACE.

?MKLCSR Command switch is required

Description: You typed an incomplete command string. Supply additional information with file switches.

Suggested User Response: Reissue the command string including the necessary switches on either the master library or transaction files.

THE MAKLIB PROGRAM

?MKLEEI .ENDE seen before .ENDI in edit

Description: Your .FIX file is missing an .ENDI pseudo-op, or the .ENDI pseudo-op is out of order.

Suggested User Response: Make sure that each .INSERT pseudo-op is matched with an .ENDI pseudo-op to identify the code to be inserted. The .ENDE pseudo-op indicates the end of an edit. Therefore, it is always the last statement of an edit.

THE MAKLIB PROGRAM

?MKLEFF End of file found before END block in module

Description: The input master file is bad. Although part of the file is readable, the END block (REL Block type 5) is missing for the particular module. This indicates that the file is damaged.

Suggested User Response: Re-create the file or restore it from a backup copy.

%MKLEMA Entire MASTER file will be appended

Description: For the current command string, the entire MASTER file is being appended to the output file even though you specified an individual module.

?MKLEPM .EDIT pseudo-op is missing from FIX file

Description: The .FIX file is incorrect.

Suggested User Response: Change the .FIX file so that each edit begins with the .EDIT pseudo-op and ends with an .ENDE pseudo-op.

?MKLERI Edit [edit] tried to .REMOVE or .REINSERT itself

Description: The .FIX file contains an edit that has a .REMOVE or .REINSERT pseudo-op referencing itself. These pseudo-ops must reference edits that have previously been applied to the module.

Suggested User Response: Change the .FIX file so that the .REMOVE or .REINSERT pseudo-op does not reference the current edit.

?MKLETC MACRO code expression too complex at [location] (Edit [edit])

Description: The expression at the indicated location is too complex for MAKLIB to evaluate.

Suggested User Response: Try to break the expression into several simpler expressions.

?MKLETL ENTRY block is too large to read in for module [module]

Description: MAKLIB does not have enough space allocated to process all the entry points for the specified module.

Suggested User Response: Try to reduce the number of entry points in the module.

?MKLFF4 Cannot apply FIX to F40 produced REL file

Description: You attempted to apply a .FIX file to a .REL file produced by the F40 FORTRAN compiler. This operation is not supported by MAKLIB. F40 generates .REL files that MAKLIB cannot edit from .FIX files.

?MKLFNI Qualifier ^F not implemented

Description: MAKLIB does not support ^F for entering a fixed-point decimal number.

Suggested User Response: Enter the value in an alternate form.

THE MAKLIB PROGRAM

?MKLFSI File status error on input [status] for file [file]

Description: An error occurred while MAKLIB was reading the specified file. The status value that appears is described in this manual in Table 5-3.

Suggested User Response: This could indicate a more global problem with the system. Refer to Table 5-3 in this manual to determine the specific problem with the named file.

?MKLFSO File status error on output [status] for file [file]

Description: An error occurred while MAKLIB was writing the specified file. The status value that appears is described in this manual in Table 5-3.

Suggested User Response: This could indicate a more global problem with the system. Refer to Table 5-3 in this manual to determine the specific problem with the named file.

?MKLIAA Illegal address in .ALTER in edit [edit]

Description: In the specified edit, you used a .ALTER pseudo-op to change the contents of an address. However, you gave an illegal value for an address.

Suggested User Response: Change the .ALTER pseudo-op so that it specifies a legal address.

?MKLIAI Illegal address in .INSERT in edit [edit]

Description: You specified an illegal address in the location field for a .INSERT pseudo-op.

Suggested User Response: See if the address you specified is the address of a literal, external, or undefined address. These are not allowed.

?MKLIAL .INSERT address is not in current module in edit [edit]

Description: You specified an address in the location field for a .INSERT pseudo-op that is not in the specified module.

Suggested User Response: First determine if the address you specified is the one you intended. Then check the .FIX file to verify that you placed the .INSERT sequences after the correct .MODULE pseudo-ops.

?MKLIBT Illegal block type ([type]) was seen in file [file]

Description: MAKLIB encountered an illegal REL Block while reading the indicated file.

Suggested User Response: The .REL file being read may be damaged. Re-create the .REL file and try again.

?MKLIED Internal error detected at [location] in MAKLIB

Description: This indicates that MAKLIB cannot perform the operation you were attempting.

Suggested User Response: Contact your Software Specialist or send a Software Performance Report (SPR) to DIGITAL.

THE MAKLIB PROGRAM

?MKLIIA .INSERT pseudo-op illegal inside range of .INSERT [edit]

Description: In the specified edit, you nested .INSERT psuedo-ops.

Suggested User Response: These pseudo-ops cannot be nested. Use the .ENDI pseudo-op to end the first .INSERT sequence before you begin another.

?MKLIII Illegal pseudo-op in range of .INSERT: [value] at [location] (edit[edit])

Description: In your .FIX file you used a MACRO pseudo-op which is illegal in a .INSERT sequence.

Suggested User Response: Verify that this is the correct MACRO pseudo-op for the operation you wish to perform.

?MKLILS Illegal use of long string or BLOCK in .ALTER at [location] (edit[edit])

Description: You tried to use a multi-word value with the .ALTER pseudo-op.

Suggested User Response: The .ALTER pseudo-op is for changing single word values only. This applies to the original value as well as the new value. Use a separate .ALTER pseudo-op for each word to be altered.

?MKLIPM .ENDI seen without .INSERT in edit [edit]

Description: The .FIX file is incorrect.

Suggested User Response: Change the .FIX file so that each set of instructions you wish to insert begins with an .INSERT pseudo-op and ends with an .ENDI pseudo-op.

?MKLIRF Illegal relocation in FORWARD reference to [symbol] in edit [edit]

Description: MAKLIB could not process the reference to the specified relocatable symbol.

?MKLIRM /INSERT requires at least one /MASTER specification

Description: The command string is incomplete.

Suggested User Response: When you specify /INSERT on a transaction file, you must include /MASTER with the master file. You must supply a specification on /MASTER for every module you wish to insert.

?MKLISM /INSERT,/REPLACE and /FIX are illegal switches on MASTER

Description: The command string is incorrect.

Suggested User Response: Reissue the command string putting /INSERT, /REPLACE, or /FIX on the transaction file. When you use /INSERT or /REPLACE, you must include /MASTER on the master file.

THE MAKLIB PROGRAM

?MKLIST Interim symbol table overflowed, Code too complex in edit [edit]

Description: Your specified edit contains code that has too many symbols for MAKLIB to process.

Suggested User Response: Try to break the single edit into several edits, each having a fewer number of symbols.

?MKLITS Insufficient TRACE block storage in edit [edit]

Description: The specified edit exhausted all allocated storage for TRACE block information.

Suggested User Response: Try to break the edit into several edits, thus reducing the amount of storage MAKLIB needs for TRACE blocks at any one time. For each edit, MAKLIB creates a TRACE block for each module that is changed.

?MKLIUN Illegal to have null address in .INSERT in edit [edit]

Description: In the specified edit, the location field of a .INSERT pseudo-op is null.

Suggested User Response: The location field cannot be null. It specifies the location where you want to insert the patch code.

%MKLLII Label outside of .INSERT was ignored: [label]

Description: In a .FIX file you specified a label outside the range of a .INSERT.

Suggested User Response: Change the .FIX file so that the label occurs within the range of an .INSERT-.ENDI pair. This specifies where you insert the new label in the program.

?MKLLTL MACRO code line is too long at [location] (Edit [edit])

Description: In the specified edit, a line of code in the .FIX file is too long for MAKLIB to process.

Suggested User Response: Try to reduce the length of the line by breaking it into several shorter lines.

?MKLMCA Pseudo-operator argument error at [location] (Edit [edit])

Description: You gave an illegal pseudo-op argument. The values you can use depend on the particular pseudo-op.

Suggested User Response: Supply a legal value for that pseudo-op.

?MKLMCB MASTER device must be capable of binary IO

Description: The input master file in your command string is not on a device that is capable of binary input/output. MAKLIB performs binary input/output on the master file.

Suggested User Response: Keep the files you plan to use as MAKLIB master files on devices capable of binary input/output.

THE MAKLIB PROGRAM

?MKLMCE Command error

Description: MAKLIB was not able to recognize a valid command from the command string that you typed. This error could occur if you improperly formatted the command string or if you used a non-unique abbreviation to specify a switch.

Suggested User Response: Retype the command string in the correct format: Destination File Spec=Source File
Spec1/Switches,Source File Spec2/Switches,...Source File
Specn/Switches

?MKLMCF Illegal forward or external reference at [location] (Edit [edit])

Description: At the specified location you made a reference to an undefined symbol in this module. It could be a forward reference to a new symbol or it could be an external reference. MAKLIB cannot process forward references to new symbols. This error could also occur if you attempted to reference an existing symbol and supplied the wrong symbol name.

Suggested User Response: Change your forward or external references to legal ones.

?MKLMCM Attempt to redefine value of symbol [symbol] at [location] (Edit [edit])

Description: You tried to redefine the value of an existing symbol. MAKLIB does not support this.

Suggested User Response: You cannot use MAKLIB to change the value of an existing symbol. If this is not what you intended to do, you may be using a symbol that was already defined. Try another symbol.

?MKLMCN MACRO code numeric error at [location] (Edit [edit])

Description: You supplied a numeric argument at the specified location that is illegal in that context.

Suggested User Response: Supply a legal numeric value.

?MKLMCQ MACRO code is questionable at [location] (Edit [edit])

Description: MAKLIB cannot process the code at the specified location.

Suggested User Response: Check the code for legal MACRO syntax.

?MKLMCR MACRO code relocation error at [location] (Edit [edit])

Description: The value at the specified location contains a half-word that is neither absolute nor simple relocatable. Expressions that would require Polish, such as A + A where A is relocatable, cannot be handled in MAKLIB.

Suggested User Response: Give absolute or simple relocatable values only.

THE MAKLIB PROGRAM

?MKLMCU Undefined symbol: [symbol] at [location] (Edit [edit])

Description: The symbol shown is not defined.

Suggested User Response: Define the symbol. If you are trying to use a symbol that you think is already defined, check for the correct symbol name.

?MKLMCW BYTE,EXP,DEC,or OCT more than one word at [location] (Edit [edit])

Description: You used an expression that generates a multi-word value. In this context, MAKLIB supports only a single word value.

Suggested User Response: Try to break the expression into simpler expressions, each generating a single word value.

?MKLMEP Missing .ENDE for edit [edit]

Description: The specified edit in your .FIX file is incorrect.

Suggested User Response: Change the .FIX file so that each edit begins with the .EDIT pseudo-op and ends with an .ENDE pseudo-op.

?MKLMFR Master file rejected by condition

Description: The master file did not meet the condition you specified; hence processing cannot continue.

?MKLMHE Module already has an edit [edit]

Description: You attempted to apply an edit to a module that already has an edit with a similar identifier.

Suggested User Response: Each edit to a module must have a unique identifier. You may have already applied this edit to the module.

?MKLMKM [Pseudo-op] pseudo-op in edit [edit] without preceding .MODULE

Description: In the specified edit, you gave a pseudo-op out of order. It applies to a specific module that you must specify with a preceding .MODULE pseudo-op.

Suggested User Response: Make certain that a .MODULE pseudo-op precedes those pseudo-ops that refer to a specific module.

?MKLMNF Module [module] was not found in file [file]

Description: MAKLIB cannot find the specified module in this file. If you are trying to apply a .FIX to a master library, you may have specified a non-existent module name on a /MODULE pseudo-op. You can also receive this error when you perform an operation that does not involve .FIX files. You may have specified several module names and supplied them out of order. The module in question may be in the file. But since MAKLIB processes files in a sequential manner, it will not find all modules.

Suggested User Response: Use /LIST to see if the specified module is really in the library. If not, you cannot perform this function.

THE MAKLIB PROGRAM

BMKLMNI /MASTER module names are ignored when patching.

Description: For editing, the correct way to specify module names is with the .MODULE pseudo-op in the .FIX file. /MASTER is not required for this type of command. MAKLIB ignores it.

THE MAKLIB PROGRAM

?MKLMTF /MASTER switch cannot be used on transaction file

Description: The command string is incorrect.

Suggested User Response: Retype the command string with the switch in the correct place. Include only /MASTER on the master file.

?MKLNEA Not enough arguments specified

Description: The command string is incomplete. This error usually means that you omitted a required file spec.

Suggested User Response: Retype the command string in the correct format: Destination file spec=Source file spec1/Switches,Source file spec2/Switches...Source file specn/Switches

?MKLNEC Not enough core is available

Description: MAKLIB cannot obtain enough memory to process this command.

Suggested User Response: Break the .REL files into smaller numbers of modules, or break large modules into smaller modules.

?MKLNEI Null argument to .EDIT is illegal

Description: You did not specify an edit identifier on the .EDIT pseudo-op.

Suggested User Response: Each .EDIT pseudo-op requires an argument (up to 6 characters) that identifies the edit.

%MKLNIO Output file [file] will not be indexed

Description: The output file will not include an index. To add an index to the file, issue a separate MAKLIB command with /INDEX.

?MKLNMS Null specification to .MODULE [edit]

Description: In your edit, you included a .MODULE pseudo-op without the module name.

Suggested User Response: Correct the .FIX file by supplying a module name on each .MODULE pseudo-op.

?MKLNPC No program code was found for module in edit [edit]

Description: You have specified a non-existent module in the master file.

?MKLNPS No program names were specified for file [file]

Description: You tried to manipulate some of the modules in the specified file, but you did not supply any module names.

Suggested User Response: In order to /EXTRACT or /DELETE modules from a file, supply the module name on the file switch.

THE MAKLIB PROGRAM

?MKLNRP Not a recognized position switch: [value]

Description: You gave an illegal position indicator on an .INSERT pseudo-op.

Suggested User Response: Use one of the three legal position indicators: BEFORE, AFTER, or REPLACE.

?MKLNTM Not enough TRANSACTION modules were specified

Description: You did not supply enough replacement module names to perform the /REPLACE operation.

Suggested User Response: When you give the MAKLIB command string, make certain that there is a corresponding replacement module for each module you intend to replace in the master library.

?MKLODD Output device must be DISK or DECTAPE

Description: In your MAKLIB command, you specified an illegal output library device.

Suggested User Response: Retype the command string and use disk for the output device of the library file.

?MKLPEF Premature end-of-file during edit [edit] in file [file]

Description: While processing the indicated edit, MAKLIB encountered an unexpected end of file in the .FIX file. This error usually occurs when you omit the .ENDE pseudo-op.

Suggested User Response: Check the .FIX file for errors, and look especially for .ENDI and .ENDE pseudo-ops.

%MKLPEP Precluded edit [edit] is present in module

Description: The module you are editing contains an active edit that your current edit precludes with the .ASSOCIATED pseudo-op. MAKLIB still applies your edit.

%MKLPES Purging EXTERNAL symbol [symbol] may give bad REL file

Description: One of the symbols that this edit PURGED from the symbol table was an EXTERNAL symbol. With this symbol removed, it may not be possible to LINK the file correctly.

?MKLRBF REQUEST or REQUIRE block is badly formatted

Description: In the master library being processed, MAKLIB encountered a REQUEST block (REL Block type 17) or REQUIRE block (REL Block type 16) that was not in the expected format.

Suggested User Response: The library file may be damaged. Try to rebuild it.

%MKLREM Required edit [edit] is missing from module

Description: The module you are editing is missing an active edit that your current edit requires with the .ASSOCIATED pseudo-op. MAKLIB still applies your current edit.

THE MAKLIB PROGRAM

%MKLRER Required edit [edit] is inactive in module

Description: The module you are editing contains an inactive edit that your current edit requires with the .ASSOCIATED pseudo-op. MAKLIB still applies your current edit.

%MKLRIA Edit [edit] tried to .REINSERT already active edit

Description: Your current edit contains a .REINSERT pseudo-op that attempts to activate an edit that is already active.

%MKLRIE Edit [edit] tried to .REMOVE already inactive edit

Description: Your current edit contains a .REMOVE pseudo-op that attempts to deactivate an edit that was previously deactivated.

%MKLRIN Edit [edit] tried to .REINSERT non-existent edit

Description: Your current edit contains a .REINSERT pseudo-op that attempts to activate a non-existent edit.

%MKLRNE Edit [edit] tried to .REMOVE non-existent edit

Description: Your current edit contains a .REMOVE pseudo-op that attempts to deactivate a non-existent edit.

?MKLRTL .INSERT's REPLACE argument of [value] too large for module [module]

Description: On an .INSERT pseudo-op you specified a number of instructions to be replaced. This number exceeds the number of instructions in the module beyond the starting replacement address.

Suggested User Response: Correct the argument to the REPLACE keyword on the .INSERT pseudo-op.

?MKLSCE Storage for patch code was exhausted in edit [edit]

Description: MAKLIB allocates a fixed amount of space for processing the new code inserted from a .FIX file. Your .FIX file contains more code than MAKLIB can process in the allocated space.

Suggested User Response: Try to break your .FIX file into several .FIX files with fewer lines of new code.

?MKLSIO Switches are illegal on output

Description: The command string is incorrect. Switches are not recognized on the output file.

Suggested User Response: Retype the command string including any necessary switches with the appropriate input files.

%MKLSNF Symbols not found for module

Description: There are no symbols in the master file for the module that you wish to edit.

THE MAKLIB PROGRAM

?MKLSSE Storage for patch symbols was exhausted during edit [edit]

Description: MAKLIB allocates a fixed amount of storage for processing the new symbols from .FIX files. Your .FIX file contains more symbols than MAKLIB can process in the allocated space.

Suggested User Response: Try to break your .FIX file into several .FIX files with fewer symbols.

%MKLTBF TRACE block is badly formatted in module

Description: One of the TRACE blocks (REL Block type 1060) for this module is not in the expected format. The .REL file may be damaged. Since the loader ignores trace blocks when reading a file, you may still be able to load from this .REL file.

%MKLTFI Transaction file ignored

Description: You included a transaction file in the command string to create an indexed library.

Suggested User Response: You must create the indexed library as a single operation. If there are several operations you must perform on the library, the command to index the library should be the last command that you issue.

?MKLTFR All transaction files rejected by condition.

Description: Processing cannot continue because the transaction files did not meet the condition you specified (as size, creation date, etc.).

?MKLTMN Too many module names . . . stopped at [module]

Description: The command string is too long.

Suggested User Response: Retype the command string as several shorter commands. It is unlikely that this message will appear, since MAKLIB now allows up to 100 switch arguments for each command string.

?MKLTMS Too many switches

Description: You included too many switches on a file spec. You specified some of the switches in an illegal combination.

Suggested User Response: Retype the command string in a correct format.

?MKLUDF Module [module] in edit [edit] contains undefined symbol(s)

Description: Your current edit contains undefined symbols in the indicated module.

Suggested User Response: Make certain that all new symbols introduced in your .FIX file have values associated with them.

?MKLWIO Wild cards illegal for output file specification

Description: You gave an output file name in the command string that included wildcard characters (either * or ?).

Suggested User Response: Since wildcard characters are illegal for the output file name, retype the command string with an explicit output file name.

THE MAKLIB PROGRAM

?SCNSVR Switch value required on [switch]

Description: The specified switch requires a value.

Suggested User Response: Retype the command string and supply a value for the switch. The format for providing a value is /switch:value or /switch:(value1,value2).

?WLDLKE Protection failure file [file]

Description: The specified file is protected. You do not have the privileges to access it.

Suggested User Response: If this is the output file, you need privileges to create a file in the directory you specified. If it is an input file, you need privileges to read that file from that particular directory.

?WLDLKE Non-existent file [file]

Description: The specified file does not exist.

Suggested User Response: This error usually occurs when the name of an input file is incorrect. Retype the command string with the correct input file name.

6.5 TECHNICAL NOTES

The following is supplementary information related to editing libraries with MAKLIB. Section 6.5.1 describes TRACE blocks (REL Block type 1060). Section 6.5.2 contains the format for code insertion in .FIX files.

6.5.1 Format of TRACE Block Data (REL Block Type 1060)

MAKLIB uses the TRACE block to include, in the .REL file, information for verifying and changing the patch status of a program. The format of the TRACE block follows.

The first part of the TRACE block is the static area. This area appears in each module affected by the particular edit. The static areas give information common to all modules affected by an edit and the variable gives the changing data on the particular edit as it goes from module to module.

| | | |
|---------|-------------------------|-------------------|
| TB\$HED | REL Block Type | Length of Block |
| TB\$EDT | SIXBIT Edit Name | (Up to 6 chars.) |
| TB\$STA | -1 If Active | Who Last Affected |
| TB\$MAK | Who Created | Date (15 BIT) |
| TB\$INS | Who Installed | Date (15 BIT) |
| TB\$FUT | Reserved for Future Use | |
| TB\$LEN | # of Assoc. Edits | # of PCO Groups |

THE MAKLIB PROGRAM

The static area, which repeats in each module, is followed by a variable area. The variable area consists of two parts. The first gives data on the associated edit status for this module, and the second gives the actual program change orders (PCO's). The length of each of these areas appears in the static area of the TRACE block.

For each associated edit, the following group appears:

| | | |
|---------|---------------------------------|-------------------------|
| TB\$AEN | SIXBIT Edit Name of Assoc. Edit | |
| TB\$AES | X | Reserved for Future Use |

0B0 If can't be present
1B0 If must be present

After the associated edit groups appear (if there are any), the PCO groups for that module appear. There are currently three types of program change groups: INSERT, REMOVE, and REINSERT. They can appear in any order and the total number is variable.

INSERT PCO:

| | | |
|---------|----------------------------|---------------------|
| TB\$PCO | PCO Type Code (1) | Length of Group |
| TB\$DAT | Instrs. INSERTed | Addr. of INSERT |
| TB\$PAT | New Addr. of Original Code | Addr. of Patch Code |

REMOVE PCO:

| | | |
|---------|-------------------|-----------------|
| TB\$PCO | PCO Type Code (2) | Length of Group |
| TB\$REN | SIXBIT Edit Name | |

RE-INSERT PCO:

| | | |
|---------|-------------------|-----------------|
| TB\$PCO | PCO Type Code (3) | Length of Group |
| TB\$RIN | SIXBIT Edit Name | |

ALTER PCO:

| | | |
|---------|----------------------------|-----------------|
| TB\$PCO | PCO Type Code (4) | Length of Group |
| TB\$DAT | Unused | Addr. of ALTER |
| TB\$PAT | New Addr. of Original Code | Unused |

THE MAKLIB PROGRAM

6.5.2 Format of Code Insertion

The four formats of code insertion in a .FIX file are shown here. Notice that, in all cases, the patch ends with exactly two JUMPA instructions. Thus, the last instruction of the patch can at most skip a single instruction and still return control to the original code.

To INSERT any instruction or series of instructions (code) BEFORE a location, use this format:

.INSERT location, BEFORE, <original instruction>

LOCATION:

JUMPA %PATCH

%PATCH:

First Patch Instruction

Second Patch Instruction

·
·
·

Last Patch Instruction

Original Patch Instruction

JUMPA 1, LOCATION+1

JUMPA 2, LOCATION +2

Any "Literals"

The actual label created at the location of the patched-in code is of the form:

"%<edit-name><edit-part>

where the edit-part is from "A" to "Z", incremented for each .INSERT in the edit.

THE MAKLIB PROGRAM

To INSERT any instruction or series of instructions (code) AFTER a location, use this format:

.INSERT location, AFTER, <original instruction>

| | |
|-----------|--------------------------|
| LOCATION: | JUMPA %PATCH |
| %PATCH: | Original Instruction |
| | First Patch Instruction |
| | Second Patch Instruction |
| | . |
| | . |
| | . |
| | Last Patch Instruction |
| | JUMPA 1, LOCATION+1 |
| | JUMPA 2, LOCATION+2 |
| | Any "Literals" |

To REPLACE a single instruction, use the format:

.INSERT location, REPLACE, <original instruction>

| | |
|-----------|------------------------------|
| LOCATION: | JUMPA %PATCH |
| %PATCH: | Original Instruction |
| | First Patch Instruction |
| | Second Patch Instruction |
| | . |
| | . |
| | . |
| | nTH (Last) Patch Instruction |
| | JUMPA 1, LOCATION+n |
| | JUMPA 2, LOCATION+n+1 |
| | Any "Literals" |

If you do not insert instructions (n=0), then the return is to LOCATION+1 and LOCATION+2.

THE MAKLIB PROGRAM

To REPLACE more than one instruction at a location, use the format:

.INSERT location REPLACE:m <original instruction>

LOCATION:

%PATCH:

| | |
|---------------------------|--------|
| JUMPA | %PATCH |
| Original Instruction | |
| First Patch Instruction | |
| Second Patch Instruction | |
| . | |
| . | |
| . | |
| Last Instruction of Patch | |
| JUMPA 1, LOCATION+m | |
| JUMPA 2, LOCATION+m+1 | |
| Any "Literals" | |

If you do not specify m, or it is zero, the effect is the same as the REPLACE keyword without an argument. In other words, one word is skipped over on return for every one inserted.

CHAPTER 7

THE DUMPER PROGRAM

7.1 INTRODUCTION

DUMPER is a TOPS-20 utility program used to save files on magnetic tape, and later to restore any or all of these files to a specified directory on disk. DUMPER is available to both nonprivileged and privileged users. (Throughout this chapter, the term "operator" is occasionally used instead of privileged user.) As a nonprivileged user, you can use DUMPER to save your own files or any files to which you have access, by transferring them from the disk area to a 9-track magnetic tape (DUMPER does not function on a 7-track magnetic tape drive), and later restoring them to disk. As a privileged user, you can use DUMPER to:

- save other users' files
- back up the system files (copy all files onto tape for an indefinite period of time)
- archive users' files (copy files marked for storage onto tape and delete them from disk)
- migrate users' files (copy files onto tape and delete them from the disk) to create added disk space
- retrieve previously migrated or archived files.

The length of time files remain on tape is determined by each installation. In general, however, archived files are kept longer than backup files. Archiving is voluntary on the part of the nonprivileged user; migrating is not.

Sections 7.2 through 7.4 describe the functions of DUMPER, the use of tapes, and the procedure to run DUMPER. Sections 7.5 and 7.6 describe the commands available to both the nonprivileged and privileged user. Examples are provided to illustrate the use of each command. Included in the examples are error messages that might occur with a particular command. Section 7.7 contains an alphabetical list of all commands. Section 7.8 contains an alphabetical list of all error messages.

THE DUMPER PROGRAM

Before continuing, you should know the following terms:

| | |
|--------------|--|
| JFN | Indicates a job file number, an octal number that represents a particular file. |
| saveset | Indicates a group of files on tape stored as the result of one SAVE command to DUMPER. |
| saveset name | Indicates a string of up to 30 alphanumeric characters used as the name for a saveset. The specified name is written in the saveset header on the tape. |
| tape set | Indicates a set of one or more volumes (reels) of tapes grouped under a single name. Each tape is distinguished by a unique identification of up to six alphanumeric characters. |

NOTE

As mentioned at the beginning of this manual, it is assumed that you are familiar with TOPS-20 log-in procedures and the basic commands. To understand DUMPER, you should be particularly familiar with the following commands: MOUNT TAPE, DISMOUNT TAPE, DEFINE, ASSIGN, and INFORMATION. (Refer to the TOPS-20 Commands Reference Manual.)

7.2 FEATURES

The following is a list of DUMPER's most useful features.

- With DUMPER, you can specify particular files to be transferred between disk and tape. For example, you can specify files using the standard TOPS-20 file spec format of dev:<dir>name.typ.gen and/or you can select files based on the dates and times that the files were created, modified, or accessed. Other conditions can also be set; if the file meets all conditions, it is transferred. (Refer to Section 7.5.1.)
- You can save a set of files that exceeds one reel of magnetic tape. If all files specified cannot fit on one tape, DUMPER continues the operation on subsequent tapes (except in INTERCHANGE mode, which allows only one tape).
- You can recover from a system crash during a save or restore operation without restarting at the beginning. (You do this by adding an additional reel of tape, and using the INITIAL command. Refer to Section 7.5.1.)

THE DUMPER PROGRAM

- As a privileged user, you can transfer files marked for archival by another user. This ensures that those files are saved for a period of time (determined by each installation) for future use or reference.
- As a privileged user, you can migrate files that have not been referenced within a specified period of time (as defined by each installation).
- As a means of verification, you can request that filenames, directory names, and saveset names be printed during save and restore operations.
- DUMPER can read and write tapes written under older versions of DUMPER. Section 7.3 describes the use of tapes with both Version 4 and previous versions of TOPS-20 DUMPER.

7.3 USING TAPES WITH AND WITHOUT TAPE DRIVE ALLOCATION

Before running DUMPER, you must have a tape mounted. With Version 4 of TOPS-20, your installation may be using tape drive allocation; with previous versions, tape drive allocation does not exist. Tape drive allocation allows you to request a tape to be mounted; TOPS-20 fulfills that request with any free drive. The absence of tape drive allocation requires that you assign a specific tape drive yourself. If you are using unlabeled tapes, tape drive allocation has no effect on the content of the tapes that DUMPER writes. If you are using labeled tapes, tape drive allocation is required. The label information on the tape then helps to identify the tape without operator intervention. Tape drive allocation also allows you to read tapes written under versions of DUMPER previous to Version 4 of TOPS-20. The difference between the presence or absence of tape drive allocation is in the method of mounting and dismounting tapes, and the fact that you cannot read or write labeled tapes without tape drive allocation. A description of each method follows.

NOTE

You can determine whether your installation is using tape drive allocation by typing the TOPS-20 command INFORMATION SYSTEM-STATUS.

@INFORMATION (ABOUT) SYSTEM-STATUS (RET)

Operator is in attendance
Remote logins allowed
Local logins allowed
Pseudo-terminal logins allowed
ARFANET terminal logins are not allowed
Console terminal login allowed
Accounting is being done
Account validation is enabled
Tape-drive allocation is enabled
Automatic file-retrieval-waits allowed
Scheduler bias-control setting is 11
Class scheduling by accounts enabled,
windfall allocated, batch class 1

@

THE DUMPER PROGRAM

If your installation does not have tape drive allocation, use a tape as follows:

- Select a tape drive (for example MTAl:) and type the TOPS-20 command ASSIGN MTAl:.
- The system responds with MTAl: ASSIGNED.
- Mount the tape on the assigned drive.
- Run the DUMPER program to perform your tape operations. First identify the assigned tape by typing the DUMPER command TAPE MTAl: in response to the prompt, DUMPER>. If you omit the command, DUMPER looks for a device that has been defined as MTA-DUMPER:. To define the logical name MTA-DUMPER: as your assigned tape drive (and thereby save a step after entering DUMPER), perform the following steps:

- Assign a tape drive.

```
@ASSIGN MTAl: (RET)
```

- Define that drive as MTA-DUMPER:

```
@DEFINE MTA-DUMPER: MTAl: (RET)
```

- Upon completion of tape operations, exit from DUMPER.
- Type the TOPS-20 commands UNLOAD MTAl: and then DEASSIGN MTAl:. Remove the tape from the drive.

If your installation is using tape drive allocation (under Version 4 of TOPS-20), you mount a tape as follows:

- Type the TOPS-20 command MOUNT TAPE name:. The name (for example, TAPE1:) is a logical name. The system responds with a message indicating that your request is in queue. For example:

```
@MOUNT TAPE TAPE1: /WRITE-ENABLED (RET)  
[Mount Request TAPE1 Queued, Request-ID 174]
```

NOTE

By typing /WRITE-ENABLED as part of the MOUNT TAPE command, you can write onto the tape. If you do not specify this switch, you can do only read operations (i.e., print a list of filenames or restore files to disk).

If your installation requires that the tape be mounted by an operator, the tape reel must have an external identification label with the specified name clearly visible to the operator.

THE DUMPER PROGRAM

NOTE

If your tape has never been initialized, be sure to tell the operator. He must initialize the tape before you can use it.

- Either you or the operator (depending upon the procedure at your installation) mounts the tape on any available drive. If the tape is an unlabeled tape, it must be identified to the system. (Refer to the TOPS-20 Operator's Guide.)
- The tape is then known to the system, and the system assigns a tape device number. You receive a message such as:

```
[Mount Request TAPE1 Queued, Request-ID 114]
[Tape set TAPE1, volume TAPE1 mounted]
[TAPE1: defined as MT0:]
```

- Run DUMPER to perform your tape operations.
- Upon completion of tape operations, exit from DUMPER.
- Type the TOPS-20 command DISMOUNT TAPE name: where the name is the same logical name used with the MOUNT TAPE command.
- The system responds with:

```
[Tape dismounted, logical name TAPE1: deleted]
```

To assign the logical name MTA-DUMPER: before entering DUMPER, define the logical name using your tape name or the tape drive number you were assigned:

```
@DEFINE MTA-DUMPER: TAPE1:RET
or
@DEFINE MTA-DUMPER: MTn:RET
```

DUMPER supports both unlabeled and TOPS-20 and ANSI labeled tapes. It is not necessary to specify which you are mounting. However, in case there is a duplication of set name or identification between labeled and unlabeled tapes, it is wise to be specific. This avoids confusion for the operator mounting your tapes. For example, you can append a label-type switch to the MOUNT TAPE command:

```
@MOUNT TAPE TAPE1: /LABEL-TYPE:UNLABELED /WRITE-ENABLEDRET
or
@MOUNT TAPE TAPE1: /LABEL-TYPE:TOPS-20 /WRITE-ENABLEDRET
```

If you think you need more than one tape, you can identify all the tapes in advance by adding another switch to the MOUNT TAPE command:

```
/VOLIDS:valid1,valid2,...,validn
```

If you use /VOLIDS:, the name you give with the MOUNT TAPE command now refers to the entire tape set. The specifications valid1 through validn (for example, DH33, DH44, DH55) identify each volume of tape within the set. Each volume identifier (valid) can be one to six alphanumeric characters.

THE DUMPER PROGRAM

When you type the MOUNT TAPE command, the system requests that the first volume of the set be mounted. When the operator has done this, you receive a message, such as:

```
[Tape set TAPE1, volume DH33 mounted]
[TAPE1: defined as MT0:]
```

If you wish to read the tape set, you must specify the volumes in the order in which they were written. You can begin with any volume in the set, but you cannot then specify a tape with a valid previous to the one with which you began. For example, if you wrote the tapes in the order DH33, DH44, DH55, you can request:

```
/VALID:DH44,DH55
```

You cannot, however, request volumes DH44, DH33 nor omit a valid by requesting volumes DH33, DH55.

When the first volume is mounted, you are ready to run DUMPER.

NOTE

If your write operations require more than the specified number of tapes, DUMPER automatically requests the operator to mount an additional tape. When the operation is complete, you can verify the additional tape in your set by typing the TOPS-20 command INFORMATION (ABOUT) VOLUMES name: before you type the DISMOUNT command (where name: is the name of the tape set). The system responds with the valid of each volume.

```
@INFORMATION (ABOUT) VOLUMES (OF TAPE) TAPE1: RET
Volumes of tape set TAPE1: DH33,DH44,DH55,DH56
@
```

7.4 RUNNING DUMPER

To run DUMPER, type DUMPER and press RETURN in response to the TOPS-20 prompt @. DUMPER prompts with its name and a right angle bracket (>):

```
@DUMPER RET
DUMPER>
```

NOTE

The examples in this chapter assume use of unlabeled tapes and tape drive allocation enabled.

As with TOPS-20 commands, you can type DUMPER commands in full, abbreviated, or recognition mode. Also, you must terminate all DUMPER commands by pressing RETURN.

THE DUMPER PROGRAM

To leave DUMPER and return to TOPS-20 command level, type either QUIT or EXIT:

```
DUMPER>EXIT RET
@
```

7.5 THE NONPRIVILEGED USER

As a nonprivileged user, you can use DUMPER to save and restore your own files, or any other files to which you have access.

The DUMPER commands to which you have access can be classified into three categories:

1. Status-Setting commands, which set parameters affecting future operation of action commands.
2. Tape-Positioning commands, which control the position of the tape.
3. Action commands, which start, stop, interrupt, or continue a file transfer or file check.

Sections 7.5.1 through 7.5.3 describe the commands in each category. Each section includes command formats and examples. Where applicable, command complements are included in the description. The complement of a command negates the command, and is formed by preceding the command with NO.

7.5.1 Setting the Status of Operation

The Status-Setting commands set parameters that affect the operation of the Action commands CHECK, RESTORE, RETRIEVE, and SAVE (refer to Section 7.5.2). Once you have set a parameter, it remains in effect until you change it or restart DUMPER. If you specify multiple conditions, the file you wish to transfer must satisfy all conditions to be transferred.

Two parameters associated with these commands are date and time. The date can be specified in any standard American or European format such as: 16-May-79, 5/16/79, etc. The time can be specified either in 24-hour format or in AM/PM format. For example 7:23 in the evening can be specified as 19:23 or as 7:23pm.

For a printed explanation of all DUMPER commands, run DUMPER and type HELP, followed by RETURN. This lists and briefly describes all DUMPER commands, and lists required arguments.

Table 7-1 lists all Status-Setting commands according to their functions. A detailed description of each command follows the table.

THE DUMPER PROGRAM

Table 7-1
Status-Setting Commands

| Function | Commands |
|--|---|
| Specifying the selection of specific files | BEFORE, ABEFORE MBEFORE, SINCE, ASINCE, MSINCE, SUPERSEDE |
| Specifying the printing (or suppressing) or storing of file or directory names | DIRECTORIES, FILES LIST, SILENCE |
| Specifying disk file characteristics | ACCOUNT, PROTECTION |
| Specifying the tape characteristics | DENSITY, FORMAT, INTERCHANGE, PARITY, TAPE, INDUSTRY, SET BLOCKING-FACTOR, SSNAME |
| Specifying checksums to be included when a list of saved files is printed | CHECKSUM |
| Specifying the continuation of interrupted savesets | SET TAPE-NUMBER, INITIAL |

The descriptions and examples below are divided into sets according to the functions described in Table 7-1. The first set describes the selection of files to be transferred on the basis of date and time, and according to name, type and generation.

The format is the same for all BEFORE and SINCE commands. Below is a description of each, followed by examples of four of the commands.

BEFORE (DATE AND TIME) date time

The BEFORE command transfers only files whose last user written date is before the date and time specified. The last user written date is the most recent time the user changed the actual data of the file. This date is recorded in the .FBWRT entry of the FDB, and is preserved when the file is copied.

NOTE

If time is omitted, the default is
00:00:01.

ABEFORE (DATE AND TIME) date time

The ABEFORE command transfers only files whose last user access is before the date and time specified. The last access date is the most recent time the file was typed, printed, or read, but not modified. This date is recorded in the .FBREF entry of the FDB.

THE DUMPER PROGRAM

MBEFORE (DATE AND TIME) date time

The MBEFORE command transfers only files whose last system write date is before the date and time specified. The last system write date is the most recent time the file was physically changed on disk. This change includes copying the file. The date is recorded in the .FBCRE entry of the FDB, and is not settable by a nonprivileged user.

NOTE

Refer to Table 7-4 for a description of FDB entries that affect DUMPER.

SINCE (DATE AND TIME) date time

The SINCE command transfers only files whose last user written date is later than the date and time specified. The last user written date is the most recent time the user changed the actual data of the file. This date is recorded in the .FBWRT entry of the FDB, and is preserved when the file is copied.

NOTE

If time is omitted, the default is 00:00:01.

ASINCE (DATE AND TIME) date time

The ASINCE command transfers only files whose last user access is later than the date and time specified. The last access date is the most recent time the file was typed, printed, or read, but not modified. This date is recorded in the .FBREF entry of the FDB.

MSINCE (DATE AND TIME) date time

The MSINCE command transfers only files whose last system write date is later than the date and time specified. The last system write date is the most recent time the file was physically changed on disk. This change includes copying the file. The date is recorded in the .FBCRE entry of the FDB, and is not settable by a nonprivileged user.

Examples:

1. DUMPER> BEFORE (DATE AND TIME) 5/18/79 8:00AM RET

transfers only files created, or whose contents were modified, before 8:00 A.M. on May 18, 1979.

2. DUMPER> ABEFORE (DATE AND TIME) 18-MAY-79 17:00 RET

transfers only files accessed by a non-write operation (i.e., those that were typed, printed, or read) before 5:00 P.M. on May 18, 1979.

THE DUMPER PROGRAM

3. DUMPER>MSINCE (DATE AND TIME) MAY-1-79 8:30 RET

transfers only files that have been system modified (for example, copied) since 8:30 A.M. on May 1, 1979.

4. DUMPER>SINCE (DATE AND TIME) 4-29-79 RET

transfers only files that have been created or changed since the first minute of April 29, 1979. (Because no time was specified, the default is one minute after midnight.)

The SUPERSEDE command sets the condition under which DUMPER overwrites a disk file with a magnetic tape file of the same name and type. You must specify one of the following conditions:

1. ALWAYS - when you want the tape file to overwrite the disk file of the same name and type regardless of the date or generation number of the disk file. If the disk file has a higher generation number, DUMPER restores the file from tape to disk, giving the disk file a generation number one higher than that existing currently. If the disk file has a lower generation number, DUMPER overwrites the disk file using the generation number of the file on tape. (Refer to the example below.)
2. NEVER - when you do not want the tape file to overwrite the disk file under any circumstances. In this case, a file is transferred to disk only if there is no file on disk with the same name and type.
3. OLDER - when the date that the tape file was last modified (created, read, written) is more recent than the date of the existing disk file with the same name and type. In that case, the tape file replaces the disk file. If the SUPERSEDE command is not specified, DUMPER assumes SUPERSEDE OLDER.

Example:

@VDIRECTORY (OF FILES) FILE1.RNO RET

MISC:<HUTCHINS>

FILE1.RNO.1#P777700 2 3155(7) 2-JUL-79 15:51:36

@DUMPER RET

DUMPER>REWIND RET

DUMPER>SUPERSEDE ALWAYS RET

DUMPER>FILES RET

DUMPER>RESTORE (TAPE FILES) FILE1.RNO RET

DUMPER tape # 1 Valid TAPE1, , Tuesday, 3-Jul-79 1126

Loading file(s) into MISC:<HUTCHINS>

MISC:<HUTCHINS>FILE1.RNO.1#P777700#A341 (TO) FILE1.RNO.3 [OK]

End of saveset

DUMPER>EXIT RET

@DIRECTORY (OF FILES) FILE1.RNO RET

MISC:<HUTCHINS>

FILE1.RNO.3

@

THE DUMPER PROGRAM

The commands to activate or deactivate printing are similar. The defaults, however, are different, as described below.

DIRECTORIES

Normally, DUMPER prints directory names on your terminal as it saves or restores them. For example,

```
@DUMPERRET
DUMPER>TAPE (DEVICE) TAPE1:RET
DUMPER>REWINDRET
DUMPER>SAVE (DISK FILES) MISC:<*>RET

DUMPER tape # 1 Valid SET2, , Tuesday, 11-Sep-79 0850
MISC:<ADLEY>
MISC:<APLB>
MISC:<APLSF>
MISC:<BAKER>
MISC:<BATOR>
MISC:<BELANGER>
*
*
*
```

To suppress printing, type NO DIRECTORIES before the SAVE or RESTORE command. This is useful if you are a privileged user performing backup of files on many directories. For example,

```
DUMPER>NO DIRECTORIESRET
DUMPER>SAVE (DISK FILES) PS:*. *.* , MISC:*. *.* (AS)RET

DUMPER tape - # 1 Valid TAPE1, , Tuesday, 3-Jul-79 1131

Total files dumped = 1143
Total pages dumped = 4293
DUMPER>
```

As all files in all directories on the structures PS: and MISC: are saved on tape, DUMPER does not print the directory names on the terminal.

To reactivate printing before the next transfer operation, type DIRECTORIES in response to the DUMPER prompt.

FILES

Normally, DUMPER does not print a list of file specs as it is saving or restoring them. You see only the name of the structure and directory, and the number of total files and pages transferred. For the file specs to print on your terminal, type FILES before the SAVE or RESTORE command.

THE DUMPER PROGRAM

Example:

```
@DUMPERRET
DUMPER>TAPE (DEVICE) TAPE1:RET
DUMPER>REWINDRET
DUMPER>DIRECTORIESRET
DUMPER>FILESRET
DUMPER>SAVE (DISK FILES) PS:*.**,MISC:*.**, (AS)RET
DUMPER tape # 1 Valid TAPE1, , Thursday, 5-Jul-79 1550
PS:<HUTCHINS>
LOGIN.CMD.5 (AS) LOGIN.CMD.5 [OK]
MAIL.TXT.1 (AS) MAIL.TXT.1 [OK]
MISC:<HUTCHINS>
ADEXMP.TST.1 (AS) ADEXMP.TST.1 [OK]
DMPMSG.RNO.1 (AS) DMPMSG.RNO.1 [OK]
DUMP.XMP.2 (AS) DUMP.XMP.2 [OK]
DUMP1.TST.1 (AS) DUMP1.TST.1 [OK]
DUMP2.TST.1 (AS) DUMP2.TST.1 [OK]
DUMP3.TST.1 (AS) DUMP3.TST.1 [OK]
DUMPER.RNO.1 (AS) DUMPER.RNO.1 [OK]
EXAMP1.TST.1 (AS) EXAMP1.TST.1 [OK]
EXAMP2.TST.1 (AS) EXAMP2.TST.1 [OK]
FILE1.RNO.6 (AS) FILE1.RNO.6 [OK]
FILE2.RNO.1 (AS) FILE2.RNO.1 [OK]
INV.RNO.2 (AS) INV.RNO.2 [OK]
PHOTO.EXE.1 (AS) PHOTO.EXE.1 [OK]
```

```
Total files dumped = 15
Total pages dumped = 125
DUMPER>
```

To deactivate printing before the next transfer operation, type NO FILES in response to the DUMPER prompt.

LIST (LOG INFORMATION ON FILE) file spec

Normally, DUMPER does not produce a file containing a list of all files as it saves them on tape. If you wish to have such a list but know it is too long to be printed at your terminal, use the LIST command, including the name of the file in which you want the list printed. For example,

```
DUMPER>LIST (LOG INFORMATION ON FILE) SPECS.LSTRET
```

When the save operation is complete, you can return to TOPS-20 command level and print the file SPECS.LST or type it on your terminal. You cannot use LIST when doing a restore.

THE DUMPER PROGRAM

Example:

```
@DUMPERRET
DUMPER>TAPE (DEVICE) TAPE1:RET
DUMPER>REWINDRET
DUMPER>LIST (LOG INFORMATION ON FILE) SPECS.LSTRET
DUMPER>SAVE (DISK FILES) *.RNORET

DUMPER tape # 1 Valid TAPE1, , Thursday, 5-Jul-79 1556
MISC:<HUTCHINS>
```

```
Total files dumped = 5
Total pages dumped = 43
DUMPER>EXITRET
@TYPE (FILE) SPECS.LST.2RET
DUMPER tape # 1 Valid TAPE1, , Thursday, 5-Jul-79 1556
```

```
Directory (number)
      file                last write          size (pages)  checksum

MISC:<HUTCHINS> (460)

      DMPMSG.RNO.1        22-Jun-79 0852        5
      DUMPER.RNO.1        29-Jun-79 1559       35
      FILE1.RNO.6         3-Jul-79 1034         1
      FILE2.RNO.1         3-Jul-79 1034         1
      INV.RNO.2           2-Jul-79 1440         1
```

Total 5 files, 43 pages

```
Total files dumped = 5
Total pages dumped = 43
@
```

If you omit a file spec with the LIST command, DUMPER automatically creates a file called LPT:DUMPER.LOG, which is printed on the line printer. If an existing file is specified, the new list of file specs is appended to the existing file.

To deactivate the LIST command before the next transfer operation, type the command NO LIST.

SILENCE

As described previously, directory names are normally printed; file specs are not. The SILENCE command suppresses the printing of directories and file specs as the files are saved or restored. The SILENCE command is therefore equivalent to NO DIRECTORIES if the FILES command has not been specified. Typing NO SILENCE is equivalent to FILES and DIRECTORIES.

The commands ACCOUNT and PROTECTION define the file characteristics for the file being restored from magnetic tape.

THE DUMPER PROGRAM

ACCOUNT (OF RESTORED FILES FROM) keyword

The ACCOUNT command specifies that any file being restored to disk assumes either your current account (specified as SYSTEM-DEFAULT) or the account of the file when it is saved on tape (specified as TAPE). Without the command, DUMPER uses TAPE.

If you specify the account incorrectly, DUMPER responds with an error message, as follows:

```
?Does not match switch or keyword
```

PROTECTION (OF RESTORED FILES FROM) keyword

The PROTECTION command specifies that any file being restored takes either the system default protection code or the protection code of the file on tape. The arguments are SYSTEM-DEFAULT (usually 777700) or TAPE. Without the command, DUMPER uses TAPE.

If you specify an invalid keyword, DUMPER responds with the same error message as for ACCOUNT.

Of the commands that set tape characteristics, the TAPE command is used most often. The others, described below, either change the basic attributes of the tape, prepare it for reading or writing with another format, or supply an identifying name for a saveset.

TAPE (DEVICE) name:

The TAPE command identifies the mounted magnetic tape to DUMPER. The name can be either the physical device assigned to you (such as MTA3: or MT0:) or logical name (such as TAPE1:). If you omit the command, DUMPER prints the reminder:

```
Tape specification needed,  
$Tape filespec
```

and waits for you to supply the device name. If you omit the device name, DUMPER responds with an error message, as follows:

```
?Invalid mastape designator, filename was not specified
```

If you type the device name incorrectly, DUMPER responds with the message:

```
?Invalid mastape designator, no such device
```

If you specify a device that is already in use by someone else, DUMPER prints:

```
?Invalid mastape designator, device already assigned to  
another job
```

As mentioned in Section 7.3, if you define your tape with the logical name MTA-DUMPER:, you need not type the TAPE command for DUMPER to access the tape.

THE DUMPER PROGRAM

DENSITY (OF MAGTAPE) n

The DENSITY command sets the density of the tape to the specified number of bits per inch (bits/in). The density can be 200, 556, 800, 1600, 6250 or JOB-DEFAULT (set by the TOPS-20 command SET TAPE DENSITY). If you do not specify the DENSITY command, DUMPER uses the density listed when you type the TOPS-20 command INFORMATION (ABOUT) TAPE-PARAMETERS. If the tape is labeled, the density is predetermined by the label information on the tape.

If you give an invalid density, DUMPER responds with an error message as follows:

?Does not match switch or keyword

PARITY (OF MAGTAPE) type

The PARITY command sets the parity of the mounted magnetic tape to EVEN or ODD. DUMPER normally uses the parity listed when you type the TOPS-20 command INFORMATION (ABOUT) TAPE-PARAMETERS (the system default is ODD). If you type anything other than EVEN or ODD, DUMPER responds with the same error message as for DENSITY.

SET BLOCKING-FACTOR (TO) n (RECORDS)

The SET BLOCKING-FACTOR command sets the number of logical records per physical record that DUMPER writes on tape. The default value is 1. The number must be in the range of 1 to 15. The maximum depends on the density of the tape as follows:

| Density | Maximum Blocking Factor |
|---------|-------------------------|
| 200 | 1 |
| 556 | 3 |
| 800 | 4 |
| 1600 | 8 |
| 6250 | 15 |

NOTE

INTERCHANGE mode must use a blocking factor of 1. If the blocking factor is changed and you specify INTERCHANGE mode, DUMPER resets the blocking factor to 1.

THE DUMPER PROGRAM

Version 4 of DUMPER is capable of blocking records when it writes a tape. Blocking is defined as writing physical records that contain more than one logical record. (Versions of DUMPER previous to Release 4 write only one logical record per physical record.) The term "blocking factor" refers to the number of logical records per physical record. The advantage of blocking is to permit DUMPER to fit more information on a tape than it could without blocking, because you are diminishing the number of gaps between records. This reduces the number of tapes and the amount of time it takes to save and restore files. However, a tape written with a blocking factor other than 1, and any labeled tape, cannot be restored by DUMPER previous to Version 4 of TOPS-20. It is not necessary to give this command when restoring files from tape; DUMPER automatically determines the blocking factor by reading the tape, and then processes the tape accordingly.

If you specify anything other than a decimal number with the SET BLOCKING-FACTOR command, DUMPER responds with an error message as follows:

```
?First nonspace character is not a digit
```

There can be only one blocking factor for each tape.

FORMAT (VERSION NUMBER IS) n

The FORMAT command allows DUMPER to write tapes that can be read by earlier versions of DUMPER. The number specified must be decimal; the default is 4. If you do not use a decimal number, DUMPER responds with an error message, as follows:

```
?NOT A DECIMAL NUMBER
```

INDUSTRY (COMPATIBLE 36-BIT MODE)

The INDUSTRY command allows DUMPER to read and write tapes using the INDUSTRY-COMPATIBLE tape data format of 36 bits. Normally, DUMPER uses CORE-DUMP format.

To return to CORE-DUMP format, type NO INDUSTRY in response to the DUMPER prompt.

INTERCHANGE (FORMAT)

The INTERCHANGE command allows DUMPER to read and write tapes written or to be read by the TOPS-10 BACKUP program. (Normally DUMPER uses its own format.) You should never use INTERCHANGE when writing tapes to be read by another TOPS-20 system. To deactivate the command, type NO INTERCHANGE. The example below shows a tape written by the TOPS-10 BACKUP program followed by an example of reading a tape using the TOPS-20 DUMPER program in INTERCHANGE mode.

THE DUMPER PROGRAM

```
.MOUNT MTB:BACKUP/WRITE:YES/REELID:MACS (RET)
[Mount Request BACKUP Queued, Request-ID 109]
[Volume MACS Mounted on MTB263 as Logical Name BACKUP]
BACKUP mounted, MTB263 used
```

```
.R BACKUP (RET)
```

```
/TAPE BACKUP
/FILES
/INTERCHANGE
/REWIND
/SAVE *.MACC30,5153]
!30,5153 DSKC
DECPNT MAC
LOCAL MAC
READ MAC
TRMTYP MAC
NDBDEF MAC
NETDPY MAC
PRVCK MAC
DET MAC
```

```
"Done
```

```
/EXIT
```

```
.DISMOUNT BACKUP: (RET)
MTB263 Dismounted
```

```
*
```

```
@
@MOUNT TAPE (NAME) MACS: /READ-ONLY /LABEL-TYPE:UNLABELED (RET)
[Mount Request MACS Queued, Request-ID 127]
[Tape set MACS, volume MACS mounted]
[MACS: defined as MT2:]
@DUMPER (RET)
DUMPER>TAPE (DEVICE) MACS: (RET)
DUMPER>INTERCHANGE (FORMAT) (RET)
DUMPER>REWIND (RET)
DUMPER>FILES (RET)
DUMPER>RESTORE (TAPE FILES) *.* (TO) DSK*:<*>*.*.* (RET)
```

```
DUMPER tape # 1 Valid MACS, , Tuesday, 18-Sep-79 1131
Loading file(s) into MISC:<HUTCHINS>
DECPNT.MAC. (TO) DECPNT.MAC.1 [OK]
LOCAL.MAC. (TO) LOCAL.MAC.1 [OK]
READ.MAC. (TO) READ.MAC.1 [OK]
TRMTYP.MAC. (TO) TRMTYP.MAC.1 [OK]
NDBDEF.MAC. (TO) NDBDEF.MAC.1 [OK]
NETDPY.MAC. (TO) NETDPY.MAC.1 [OK]
PRVCK.MAC. (TO) PRVCK.MAC.1 [OK]
DET.MAC. (TO) DET.MAC.1 [OK]
```

```
End of saveset
DUMPER>EXIT (RET)
@
```

THE DUMPER PROGRAM

SSNAME name

The SSNAME command specifies the name to be written in the saveset header on the tape or to appear when you type a RESTORE, SAVE, SKIP, EOT, or PRINT command. The name may contain up to 30 alphanumeric characters, including spaces. Whenever you save or restore files, the saveset name prints on your terminal.

NOTE

Unlike other commands requiring an argument, if the name is omitted, DUMPER ignores the command and uses the existing name if one is present; otherwise, it leaves the field blank. There is no error message.

The CHECKSUM command is used in conjunction with the DUMPER SAVE, and PRINT commands. (Refer to Section 7.5.2.)

CHECKSUM (FILES) type

While files are being saved, a checksum is computed. When you type PRINT, each filename is listed with a six-digit octal checksum number.

One of two types can be specified: SEQUENTIAL or BY-PAGES. If you specify SEQUENTIAL, DUMPER computes a checksum for the entire file. (In INTERCHANGE mode, you should use only sequential checksums.) If you specify BY-PAGES, the checksum includes each word of existing pages up to the End-Of-File (EOF) pointer. When you type PRINT after requesting a checksum by pages, the checksum number is followed by the letter P, as shown in the example below.

```
@DUMPER<RET>
DUMPER>TAPE (DEVICE) TAPE1:<RET>
DUMPER>REWIND<RET>
DUMPER>CHECKSUM (FILES) BY-PAGES<RET>
DUMPER>PRINT (DIRECTORY OF TAPE ONTO FILE)<RET>
```

```
DUMPER tape # 1 Valid TAPE1, , Thursday, 5-Jul-79 1555
file                                last write      size (pages)  checksum
```

```
MISC:<HUTCHINS>PHOTO.EXE.1      5-Jul-79 0830      6              442725 P
```

```
DUMPER tape # 1 Valid TAPE1, , Thursday, 5-Jul-79 1556
file                                last write      size (pages)  checksum
```

```
MISC:<HUTCHINS>DMPSG.RNO.1      22-Jun-79 0852      5              202247 P
MISC:<HUTCHINS>DUMPER.RNO.1      29-Jun-79 1559     35              677675 P
MISC:<HUTCHINS>FILE1.RNO.6       3-Jul-79 1034      1              756473 P
MISC:<HUTCHINS>FILE2.RNO.1       3-Jul-79 1034      1              756473 P
MISC:<HUTCHINS>INV.RNO.2        2-Jul-79 1440      1              565436 P
```

End of tape.

^L

THE DUMPER PROGRAM

When the checksum of the disk file disagrees with the checksum of the tape file, there is an error in the file transfer.

If you specify an invalid argument, DUMPER responds with an error message, as follows:

```
?Does not match switch or keyword
```

Two other Status-Setting commands allow you to recover after a system crash without restarting at the beginning of your save or restore operation, or to continue a save onto a second reel of tape.

INITIAL (FILESPEC) file spec

The INITIAL command allows you to continue on a new tape the operation of checking, restoring, or saving at the specified file. This command is useful in conjunction with the SET TAPE-NUMBER command if the system crashes while you are performing a check, save, or restore. For example, if DUMPER is in the process of saving FILE3 when the system crashes, mount a new tape, run DUMPER, and type

```
DUMPER>INITIAL (FILESPEC) FILE3 RET
```

In case FILE3 is distorted on the first tape, it is saved again in its entirety on the second tape.

NOTE

If you do not specify a filename, the operation begins with the first file specified in the SAVE command.

SET TAPE-NUMBER (DECIMAL NUMBER) n

The SET TAPE-NUMBER command is used for multi-reel savesets in either of two cases: when continuing a save or restore after a system crash, or when restoring files nonsequentially from a multi-reel saveset.

THE DUMPER PROGRAM

Example:

To continue a save following a crash:

```
@DUMPER(RET)
DUMPER>TAPE (DEVICE) TAPE2:(RET)
DUMPER>REWIND(RET)
DUMPER>INITIAL (FILESPEC) <HUTCHINS>FILE3.TST(RET)
DUMPER>SET TAPE-NUMBER (DECIMAL NUMBER) 2(RET)
DUMPER>SAVE (DISK FILES) *.TST(RET)
```

If you omit the tape number, DUMPER responds with an error message as follows:

?First nonspace character is not a digit

If you include a character other than a digit, DUMPER responds with either:

?First nonspace character is not a digit

or

?NOT CONFIRMED

7.5.2 Positioning the Tape

Tape-Positioning commands control the position of the tape without transferring information between the tape and disk. These commands affect the tape you specified with your last TAPE command to DUMPER. If you did not use a TAPE command, DUMPER issues a request for one, or uses a tape assigned to the logical name MTA-DUMPER:. If you are working on a system without tape drive allocation, before entering DUMPER, you must assign the tape drive to your job, using the TOPS-20 command ASSIGN.

Table 7-2 lists the four Tape-Positioning commands and their functions. Following the table is a detailed description, including examples, of each command.

Table 7-2
Tape-Positioning Commands

| Function | Commands |
|--|----------|
| Positioning to the end of the last saveset | EOT |
| Positioning to the beginning of the currently mounted tape | REWIND |
| Positioning to the end of the nth from the current saveset | SKIP n |
| Rewinding the tape entirely onto the source reel | UNLOAD |

THE DUMPER PROGRAM

The three commands below position the tape at specific points.

EOT

The EOT (End-Of-Tape) command positions the tape at the end of the last saveset and DUMPER prints the message:

Tape positioned after last saveset

As DUMPER encounters each saveset between its current position and EOT, it prints the saveset name (but not its contents) and the time and date the saveset was written.

Example:

```
@ DUMPER(RET)
DUMPER> TAPE (DEVICE) TAPE1:(RET)
DUMPER> REWIND(RET)
DUMPER> EOT(RET)

DUMPER tape # 1 Valid TAPE1, , Tuesday, 5-Jun-79 1142
DUMPER tape # 1 Valid TAPE1, , Tuesday, 5-Jun-79 1159
DUMPER tape # 1 Valid TAPE1, , Tuesday, 5-Jun-79 1327
Tape positioned after last saveset
DUMPER>
```

REWIND

The REWIND command rewinds the currently mounted volume to the beginning of the tape.

If you are using a multi-reel tape set mounted by typing the TOPS-20 command MOUNT TAPE, you can switch to a different reel by typing:

REWIND SWITCHING (TO VOLUME NUMBER) n(RET)

With this command, n specifies the number of the reel within the set; not the volume id. When you type the REWIND SWITCHING command, DUMPER releases the currently mounted tape and requests the nth volume to be mounted. (If you are using labeled tapes, DUMPER allows you to switch to volume 1 only.) The following example shows a switch to the volume identified as TWO from volume ONE:

```
@ MOUNT TAPE TAPE1: /WRITE-ENABLED /VOLIDS:ONE,TWO,THREE(RET)
[Mount Request TAPE1 Queued, Request-ID 174]
[Tape set TAPE1, volume ONE mounted]
[TAPE1: defined as MT0:]
@ DUMPER(RET)
DUMPER> TAPE (DEVICE) TAPE1:(RET)
DUMPER> REWIND SWITCHING (TO VOLUME NUMBER) 2(RET)
DUMPER>
```

THE DUMPER PROGRAM

NOTE

If you plan to switch back and forth between volumes of a set, specify the /NOUNLOAD switch with the TOPS-20 command MOUNT TAPE. This is a request to the operator not to unload any of the tapes within the set, but to use additional tape drives so that more than one tape can be mounted at a time.

SKIP (NUMBER OF SAVESETS) n

The SKIP command moves the tape over the specified number of savesets. This ensures that those savesets are not deleted by another save. DUMPER prints the name of each saveset encountered, and positions the tape at the end of the nth saveset.

If n is 0, the tape is positioned at the beginning of the current saveset. If n is a negative number, the tape backspaces by n savesets and is positioned at the beginning of that saveset. If n exceeds the total number of savesets on tape, DUMPER performs an EOT and prints the message:

%End of tape encountered

If -n exceeds the total number of savesets on tape, DUMPER reaches the beginning of the tape and prints the message:

%BEGINNING OF TAPE ENCOUNTERED

Examples:

```
@DUMPERRET  
DUMPER>TAPE (DEVICE) TAPE1:RET  
DUMPER>REWINDRET  
DUMPER>SKIP (NUMBER OF SAVESETS) 1RET
```

DUMPER tape # 1 Valid TAPE1, , Tuesday, 5-Jun-79 1142

DUMPER tape # 1 Valid TAPE1, , Tuesday, 5-Jun-79 1159
DUMPER>

```
DUMPER>SKIP (NUMBER OF SAVESETS) -1RET
```

DUMPER tape # 1 Valid TAPE1, , Tuesday, 5-Jun-79 1142
%BEGINNING OF TAPE ENCOUNTERED
DUMPER>

THE DUMPER PROGRAM

The following command releases your mounted tape from the drive.

UNLOAD

If your installation is not using tape drive allocation, you can remove your tape with UNLOAD to DUMPER.

Example:

```
DUMPER>UNLOAD RET
```

This command, however, does not deassign the tape drive.

7.5.3 Interacting with Tape Files

The Action commands affect the files on the tape specified by the last TAPE command. These commands allow you to start, stop, interrupt, or continue a file transfer or file check.

The transfer commands take, as an optional argument, one or more file specs. The file specs can contain wildcard characters. With transfer commands (SAVE and RESTORE), you can indicate both input and output (source and destination) file specs for each file specified. This allows the files to be renamed as they are saved or restored. If no destination specification is given, the specified files are transferred without being renamed. If no argument is given with the transfer commands, all files on your connected directory are saved onto tape, or all files that were saved from your connected directory, and exist in the current saveset, are restored to your connected directory.

Both the SAVE and RESTORE commands can take the source and destination arguments. In the case of SAVE, the source is the file spec identifying the name and location of the file to be saved; the destination identifies the filename under which you want the file to be stored on tape. In the case of RESTORE, the source is a file spec identifying the file you want copied from the tape; the destination is a file spec identifying the name of the file into which you want the file copied on disk.

With both commands, if you omit the source, DUMPER restores all files in the saveset corresponding to your connected directory, or saves all files as your connected directory. If you omit the destination with SAVE, DUMPER transfers the file(s) with the same file spec(s) as the one(s) on disk. If you omit the destination with RESTORE, DUMPER transfers the file(s) with the same file spec(s) as the one(s) on tape.

Table 7-3 lists all Action commands according to their functions. A detailed description of each command follows the table.

THE DUMPER PROGRAM

Table 7-3
Action Commands

| Function | Commands |
|--|------------------------|
| Transferring and comparing disk and tape files | RESTORE, SAVE CHECK |
| Printing a list of filenames on tape | PRINT |
| Halting and continuing command operation | CTRL/E, CONTINUE |
| Exiting to TOPS-20 | EXIT, QUIT |

SAVE (DISK FILES) source (AS) destination, source (AS) destination

The SAVE command creates a saveset on tape, containing all the source files specified. To rename the files as you save them, you must specify source files followed by a space, followed by destination files. With either source or destination files you can use * and % wildcard characters. To transfer additional source and destination file specs, you must separate each pair with a comma.

If you do not specify destination, DUMPER saves all source file specs under their original specifications. If you specify neither source nor destination, DUMPER saves all files on your connected directory into the saveset on tape (i.e., DUMPER assumes the file spec *.*.*).

Examples:

Note that in these examples, three savesets are created; one for each SAVE command.

```

@DUMPER RET
DUMPER>TAPE (DEVICE) TAPE1: RET
DUMPER>REWIND RET
DUMPER>FILES RET
DUMPER>SAVE (DISK FILES) DMPMSG.RNO.* (AS) DMPMSG.BAK RET

DUMPER tape # 1 Valid TAPE1, , Friday, 6-Jul-79 1145
MISC:<HUTCHINS>
DMPMSG.RNO.1 (AS) DMPMSG.BAK.1 [OK]

Total files dumped = 1
Total pages dumped = 5
DUMPER>SAVE (DISK FILES ),RNO RET

```

THE DUMPER PROGRAM

```
DUMPER tape # 1  Valid TAPE1, , Friday,  6-Jul-79 1147
MISC:<HUTCHINS>
DMPMSG.RNO.1 (AS) DMPMSG.RNO.1  [OK]
FILE1.RNO.6 (AS) FILE1.RNO.6    [OK]
FILE2.RNO.1 (AS) FILE2.RNO.1    [OK]
INV.RNO.2 (AS) INV.RNO.2        [OK]
```

```
Total files dumped = 4
Total pages dumped = 8
DUMPER>
```

DUMPER>SAVE (DISK FILES) RET

```
DUMPER tape # 1  Valid TAPE1, , Wednesday,  6-Jul-79 1217
MISC:<HUTCHINS>
ADEXMP.TST.1 (AS) ADEXMP.TST.1  [OK]
BACKER.DOR.1 (AS) BACKER.DOR.1  [OK]
DMPMSG.RNO.1 (AS) DMPMSG.RNO.1  [OK]
DUMP.XMP.1 (AS) DUMP.XMP.1       [OK]
DUMP1.TST.1 (AS) DUMP1.TST.1     [OK]
DUMP2.TST.1 (AS) DUMP2.TST.1     [OK]
DUMPER.TST.1 (AS) DUMPER.TST.1   [OK]
EXAMP1.DMP.1 (AS) EXAMP1.DMP.1   [OK]
FILE1.RNO.6 (AS) FILE1.RNO.6     [OK]
FILE2.RNO.1 (AS) FILE2.RNO.1     [OK]
INV.RNO.2 (AS) INV.RNO.2         [OK]
INV.TST.1 (AS) INV.TST.1         [OK]
PHOTO.EXE.1 (AS) PHOTO.EXE.1     [OK]
SPECS.LST.1 (AS) SPECS.LST.1     [OK]
```

```
Total files dumped = 14
Total pages dumped = 59
DUMPER>
```

NOTE

Remember that if the FILES command is not typed, the file specifications are not listed as they are saved.

If all files cannot fit on one tape, DUMPER requests that you mount a second tape. This tape is the next volume of the tape set if you type /VOLIDS: with the MOUNT TAPE command. If you do not specify the switch, the operator mounts a tape and it becomes the second volume of the set. When the additional tape is mounted, DUMPER prints a header line and resumes the save operation. (Refer to Example 9 in Section 7.5.5.)

It is important to note that savesets are stored on tape according to the position of the tape when the SAVE command is typed. If you position the tape at the beginning (refer to the REWIND command), the specified files are written there regardless of any previous files saved. If more than one saveset exists on the tape, they are all deleted when a new one is saved at the beginning of the tape. To avoid deleting existing savesets, position the tape at the end of the last saveset (EOT) or skip the number of savesets (SKIP n) you want to be sure to preserve.

THE DUMPER PROGRAM

Furthermore, if you exit from DUMPER (EXIT, or QUIT) after a save without rewinding the tape, DUMPER remains positioned at the end of that saveset. When you reenter DUMPER and type another SAVE command, the first saveset is preserved. However, DUMPER does not remember the previous record numbers or blocking factor.

NOTE

DUMPER assigns record numbers for its own use. Each time DUMPER writes a new record, it increments the record by one. With the numbering system, DUMPER can determine which records are duplicates and which have been skipped during a read operation. (There is one record number for each logical record. Therefore, if you have a blocking factor of 3, the first physical record contains records 1, 2, and 3. Refer to the SET BLOCKING-FACTOR command.)

Therefore, if you reenter DUMPER after one save, DUMPER saves new files as record 1, record 2, etc. with a blocking factor of 1. If you originally specified a blocking factor other than 1, and if you wish your files to be numbered consecutively, beginning with the first saveset, you must type REWIND, followed by EOT when you reenter DUMPER. By returning to the beginning of the tape, DUMPER determines both the blocking factor and the record numbers already saved. By then being positioned at the end of the last saveset, DUMPER preserves all existing savesets. It then continues saving files with consecutive record numbers and a blocking factor corresponding to that of the existing savesets.

RESTORE (TAPE FILES) source (TO) destination, source (TO) destination

The RESTORE command transfers the specified source files from magnetic tape to disk. To rename the files as you restore them, you must specify destination file specs. With source files, you may use the * and % wildcard characters; with destination files, you can use only the * wildcard.

If you do not specify destination, all specified files are restored with the same name, type, and generation. If you do not specify either source or destination specifications, all files saved from your connected structure and connected directory are restored to your connected structure and directory with the same name, type and generation number.

NOTE

You cannot restore files from a labeled tape under versions of TOPS-20 previous to Version 4.

When DUMPER begins restoring files to a directory, it prints the message:

Loading File(s) into <directory>

THE DUMPER PROGRAM

If you do not see this message, DUMPER is not transferring the files. The files may already exist on disk, they may have been saved from a structure or directory other than your currently connected directory, they do not exist in the current saveset, or you used a wildcard expression that does not match the files on tape.

If you are restoring files that were saved from a directory other than your own, you must specify that directory in the source file spec. If files exist on that directory with the same name and type as those on tape, DUMPER overwrites the disk files according to the condition specified with the SUPERSEDE command: ALWAYS, NEVER, or OLDER. If the SUPERSEDE command has not been typed, DUMPER assumes SUPERSEDE OLDER and overwrites the disk file only if the tape file is newer.

As is true of the SAVE command, files are restored from the current position on tape. If you want to omit one or more savesets, position the tape to the end of the last saveset to be omitted (SKIP n).

Example:

```
@DUMPER RET
DUMPER>TAPE (DEVICE) TAPE1: RET
DUMPER>REWIND RET
DUMPER>SKIP (NUMBER OF SAVESETS) 2 RET

DUMPER tape # 1 Valid TAPE1, , Tuesday, 5-Jun-79 1556

DUMPER tape # 1 Valid TAPE1, , Wednesday, 27-Jun-79 1432

DUMPER tape # 1 Valid TAPE1, , Tuesday, 3-Jul-79 1150
DUMPER>FILES RET
DUMPER>RESTORE (TAPE FILES) *.TST RET

DUMPER tape # 1 Valid TAPE1, , Tuesday, 3-Jul-79 1150
Loading file(s) into MISC:<HUTCHINS>
MISC:<HUTCHINS>DUMP1.TST.2;P777700;A341 (TO) DUMP1.TST.2 [OK]
MISC:<HUTCHINS>DUMP2.TST.2;P777700;A341 (TO) DUMP2.TST.2 [OK]
MISC:<HUTCHINS>DUMP3.TST.2;P777700;A341 (TO) DUMP3.TST.2 [OK]

End of saveset
DUMPER>
```

Once the files have been saved or restored, you can check the disk and tape versions to verify that the transfer was accurate.

CHECK (ALL TAPE FILES)

The CHECK command checks the files in the current saveset to make sure they agree (i.e., their contents and File Descriptor Blocks match) with the files on disk. If any files do not agree, DUMPER prints one or more error messages. Table 7-4 is a list of the possible differences in the FDBs that DUMPER checks.

NOTE

Be sure to position the tape at the beginning of the saveset before typing CHECK.

THE DUMPER PROGRAM

Table 7-4
File Descriptor Block (FDB) Entries Checked by DUMPER

| A difference in location | Means the files do not have the same: |
|--------------------------|---|
| .FBCTL | -Temporary, permanent, not-to-be-saved-by DUMPER, or file-class status. |
| .FBPRT | -File access code. |
| .FBCRE | -Date and time of last write to the file. It is modified when any program writes to the file. This word is changeable by a user with privileged capabilities. |
| .FBAUT | -Pointer to the string containing the name of the author. |
| .FBGEN | -Generation and directory numbers of the file. |
| .FBACT | -Account information. |
| .FBBYV | -Number of generations to retain, byte size, mode of the last write, or the number of pages. |
| .FBSIZ | -Number of bytes in the file. This word is changeable by a user with write access. |
| .FBCRV | -Date and time of creation of the file. This word is changeable by a user with write access. |
| .FBWRT | -Date and time of the last user write to the file. This word is changeable by a user with write access. |
| .FBREF | -Date and time of the last non-write access to the file. This word is changeable by a user with write access. |
| .FBCNT | -Count of writes or references. |
| .FBUSW | -Contents of the user-settable data area. |
| .FBLWR | -Pointer to the string containing the name of the user who last wrote to the file. |

Example:

```
DUMPER>REWIND (RET)
DUMPER>CHECK (ALL TAPE FILES) (RET)
```

```
DUMPER tape # 1 Valid ARCH1, , Friday, 8-Jun-79 1041
%Compare error, page 17, file DUMPER.RNO.3
%Difference in .FBCRE of file DUMPER.RNO.3
%Difference in .FBCRV of file DUMPER.RNO.3
%Difference in .FBWRT of file DUMPER.RNO.3
```

```
End of saveset
DUMPER>
```

If a file is renamed during a save or restore, using CHECK produces the message:

```
%Cannot set JFN for file [file] because:
File not found
```


THE DUMPER PROGRAM

If a disk file is modified or renamed before you do a check, the tape and disk file will not agree in either name, type, or generation. In this case, DUMPER prints the message:

```
%Cannot set JFN for file [file] because:  
File not found
```

If you want a printed list of all files saved on tape, position the tape to the beginning and type the PRINT command. This command differs from LIST in that it creates a file of filenames already on tape. LIST creates a file of filenames as DUMPER is saving the files.

PRINT (DIRECTORY OF TAPE ONTO FILE) file spec

The PRINT command records the file specs of the entire tape (beginning with the current saveset) in the specified output file. You can subsequently use the TOPS-20 TYPE or PRINT command to examine this file. The filenames are not printed on your terminal unless you specify TTY: as the output device.

```
@ DUMPER(RET)  
DUMPER> TAPE (DEVICE) TAPE1:(RET)  
DUMPER> REWIND(RET)  
DUMPER> PRINT (DIRECTORY OF TAPE ONTO FILE) FILES.LST.1(RET)  
  
DUMPER tape # 1 Valid TAPE1, , Tuesday, 5-Jun-79 1556  
  
DUMPER tape # 1 Valid TAPE1, , Tuesday, 10-Jul-79 1457  
  
DUMPER tape # 1 Valid TAPE1, , Wednesday, 11-Jul-79 1329  
  
End of tape.  
DUMPER> EXIT(RET)  
@TYPE (FILE) FILES.LST(RET)
```

If you omit the destination file spec, the contents of the current saveset are printed on your terminal.

If necessary, you may interrupt the operation of an action or Tape-Positioning command. You may then continue the operation, or cancel it by typing another command.

CTRL/E

You type the CTRL/E command by pressing the CTRL and E keys simultaneously. This command halts the action of a SAVE, RESTORE, CHECK, RETRIEVE (if you are a privileged user), or any Tape-Positioning command. When DUMPER processes the CTRL/E command, it interrupts the current operation and prints:

```
(CTRL/E)  
Interrupting...  
DUMPER>
```

THE DUMPER PROGRAM

You can type any Status-Setting command in response to the prompt, and then type CONTINUE to continue the operation at the point at which it was interrupted. If you type any other command, DUMPER asks:

%Do you really want to abort your interrupted command?

If you respond with YES, DUMPER processes the last command you typed. If you respond with NO, DUMPER gives you its prompt, allowing you to continue the interrupted command.

CONTINUE

The CONTINUE command allows you to continue the operation of an interrupted Action or Tape-Positioning command.

You cannot use the CONTINUE command if you have not interrupted DUMPER with CTRL/E, or if you have typed anything other than a Status-Setting command and have answered YES to the question:

%Do you really want to abort your interrupted command?

If you try, DUMPER responds with the message:

```
?CANNOT CONTINUE
DUMPER>
```

Example:

```
@DUMPER(RET)
DUMPER>TAPE (DEVICE) TAPE1:(RET)
DUMPER>REWIND(RET)
DUMPER>PRINT (DIRECTORY OF TAPE ONTO FILE)(RET)

DUMPER tape # 1  Valid TAPE1, , Thursday, 5-Jul-79 1550
      file                                last write      size (pages) checksum

      PS:<HUTCHINS>LOGIN.CMD.5      12-Jun-79 1147  1      052026 F
      PS:<HUTCHINS>MAIL.TXT.1(CTRL/E)

Interrupting...
DUMPER>REWIND(RET)
%Do you really want to abort your interrupted command? Y(RET)
DUMPER>CONTINUE(RET)
?CANNOT CONTINUE
DUMPER>
```

There are two commands to exit from the DUMPER program and return to TOPS-20 command level. The EXIT and QUIT commands have identical functions.

Example:

```
DUMPER>EXIT(RET)
@
```

THE DUMPER PROGRAM

7.5.4 Marking Files to be Archived

If you wish to save any of your files on the installation's archive tape, you can mark the files for archival. To do this, use the TOPS-20 ARCHIVE command. Once the operator has processed your request, only the File Descriptor Blocks (FDB's) remain. The filenames become invisible (i.e., if you take a directory of your disk area, the files appear to be gone). This is to assure you that your files have been archived and that you can no longer access or modify them. When the files have been archived successfully, DUMPER sends you a message.

NOTE

To see your files listed in the archive request queue, type the TOPS-20 command INFORMATION ARCHIVE-STATUS.

7.5.5 Additional Examples

The following examples show various ways to save and restore files, and how to recover after a system crash. The examples also demonstrate the use of Status-Setting commands before transferring files.

Examples of SAVE:

1. Mounting a tape and saving your entire directory.

Give the MOUNT TAPE command specifying /WRITE-ENABLED. Once the tape is mounted, run DUMPER, naming the saveset FULL SAVE and requesting DUMPER to list all file specifications as it saves them.

```
@MOUNT TAPE TAPE1: /LABEL-TYPE:UNLABELED /WRITE-ENABLED (RET)
[Mount Request TAPE1 Queued, Request-ID 70]
[Tape set TAPE1, volume TAPE1 mounted]
[TAPE1: defined as MT0:]
@dumper (RET)
DUMPER>TAPE (DEVICE) TAPE1: (RET)
DUMPER>REWIND (RET)
DUMPER>SSNAME FULL SAVE (RET)
DUMPER>FILES (RET)
DUMPER>SAVE (DISK FILES) (RET)

DUMPER tape # 1 Valid TAPE1, FULL SAVE, Tuesday, 3-Jul-79 1126
MISC:<HUTCHINS>
ADEXMP.TST.1 (AS) ADEXMP.TST.1 [OK]
BACKER.DOR.1 (AS) BACKER.DOR.1 [OK]
DMPMSG.RNO.1 (AS) DMPMSG.RNO.1 [OK]
DUMP.XMP.1 (AS) DUMP.XMP.1 [OK]
DUMP1.TST.1 (AS) DUMP1.TST.1 [OK]
DUMP2.TST.1 (AS) DUMP2.TST.1 [OK]
DUMP3.TST.1 (AS) DUMP3.TST.1 [OK]
DUMP4.TST.1 (AS) DUMP4.TST.1 [OK]
DUMP5.TST.1 (AS) DUMP5.TST.1 [OK]
```

THE DUMPER PROGRAM

```
DUMPER.TST.1 (AS) DUMPER.TST.1 [OK]
EXAMP1.TST.1 (AS) EXAMP1.TST.1 [OK]
EXAMP2.TST.1 (AS) EXAMP2.TST.1 [OK]
FILE1.RNO.1 (AS) FILE1.RNO.1 [OK]
FILE2.RNO.1 (AS) FILE2.RNO.1 [OK]
INV.RNO.2 (AS) INV.RNO.2 [OK]
INV.TST.1 (AS) INV.TST.1 [OK]
PHOTO.EXE.1 (AS) PHOTO.EXE.1 [OK]
```

```
Total files dumped = 17
Total pages dumped = 54
DUMPER>
```

2. Saving a single file into a second saveset.

Without rewinding the tape, type the SAVE command for one file.

```
DUMPER>SAVE (DISK FILES) PHOTO.EXE.* (AS) RET
```

```
DUMPER tape # 1 Valid TAPE1, , Wednesday, 11-Jul-79 1329
MISC:<HUTCHINS>
PHOTO.EXE.1 (AS) PHOTO.EXE.1 [OK]
```

```
Total files dumped = 1
Total pages dumped = 6
DUMPER>
```

3. Saving files from another structure.

Save all files from a mountable structure (for example, MISC:)

```
@DUMPER RET
DUMPER>TAPE (DEVICE) TAPE1: RET
DUMPER>REWIND RET
DUMPER>SSNAME MISC FILES RET
DUMPER>FILES RET
DUMPER>SAVE (DISK FILES) MISC: RET
```

```
DUMPER tape # 1 Valid TAPE1, MISC FILES, Tuesday, 3-Jul-79 1154
MISC:<HUTCHINS>
FILE1.TXT.1 (AS) FILE1.TXT.1 [OK]
FILE2.TXT.1 (AS) FILE2.TXT.1 [OK]
FILE3.TXT.1 (AS) FILE3.TXT.1 [OK]
```

```
Total files dumped = 3
Total pages dumped = 17
DUMPER>
```


THE DUMPER PROGRAM

4. Saving multiple files.

Save three separate files and rename the third one as it is saved.

```
@DUMPERRET
DUMPER>TAPE (DEVICE) TAPE1:RET
DUMPER>REWINDRET
DUMPER>FILESRET
DUMPER>SAVE (DISK FILES) FILE1.RNO,FILE2.RNO,PHOTO.EXE.* (AS) EX.ONERET

DUMPER tape # 1 Valid TAPE1, , Monday, 2-Jul-79 0829
MISC:<HUTCHINS>
FILE1.RNO.6 (AS) FILE1.RNO.6 [OK]
MISC:<HUTCHINS>
FILE2.RNO.1 (AS) FILE2.RNO.1 [OK]
MISC:<HUTCHINS>
PHOTO.EXE.1 (AS) EX.ONE.1 [OK]

Total files dumped = 3
Total pages dumped = 8
DUMPER>
```

5. Saving files modified since, or accessed before a specified time.

Save all files created or modified since 31-May-79 (00:00:01).

Note that when you do not specify a saveset name (SSNAME), two consecutive commas are printed. The saveset name normally appears between these two commas.

```
@DUMPERRET
DUMPER>TAPE (DEVICE) TAPE1:RET
DUMPER>REWINDRET
DUMPER>FILESRET
DUMPER>SINCE (DATE AND TIME) 31-MAY-79RET
DUMPER>SAVE (DISK FILES)RET

DUMPER tape # 1 Valid TAPE1, , Friday, 29-Jun-79 1156
MISC:<HUTCHINS>
ADEXMP.TST.1 (AS) ADEXMP.TST.1 [OK]
BACKER.DOR.1 (AS) BACKER.DOR.1 [OK]
DMPMSG.RNO.1 (AS) DMPMSG.RNO.1 [OK]
DUMP.XMP.1 (AS) DUMP.XMP.1 [OK]
DUMPER.TST.1 (AS) DUMPER.TST.1 [OK]
EXAMP1.DMP.1 (AS) EXAMP1.DMP.1 [OK]
EXAMP2.DMP.1 (AS) EXAMP2.DMP.1 [OK]
FILES.LST.1 (AS) FILES.LST.1 [OK]
INV.RNO.2 (AS) INV.RNO.2 [OK]
SPECS.LST.1 (AS) SPECS.LST.1 [OK]

Total files dumped = 10
Total pages dumped = 47
DUMPER>
```

THE DUMPER PROGRAM

Save files with type .QNO accessed (read in some way but not modified) before 15-July-79 at 5:00 p.m.

```
@DUMPER(RET)
DUMPER>TAPE (DEVICE) TAPE1:(RET)
DUMPER>REWIND(RET)
DUMPER>FILES(RET)
DUMPER>ABEFORE (DATE AND TIME) 7/15/79 17:00(RET)
DUMPER>SAVE (DISK FILES) *.QNO(RET)

DUMPER tape # 1 Valid TAPE1, , Wednesday, 13-Jun-79 1158
MISC:<HUTCHINS>
FILE1.QNO.3 (AS) FILE1.QNO.3 [OK]

Total files dumped = 1
Total pages dumped = 1
DUMPER>
```

6. Specifying and verifying a checksum.

Instruct DUMPER to calculate, sequentially, the checksum of all files in all savesets. When you type PRINT, DUMPER prints the file specs, including the checksum.

```
@DUMPER(RET)
DUMPER>TAPE (DEVICE) TAPE1:(RET)
DUMPER>REWIND(RET)
DUMPER>CHECKSUM (FILES) SEQUENTIAL(RET)
DUMPER>PRINT (DIRECTORY OF TAPE ONTO FILE)(RET)

DUMPER tape # 1 Valid TAPE1, , Thursday, 5-Jul-79 1555
  file                                last write      size (pages)  checksum
  MISC:<HUTCHINS>PHOTO.EXE.1          5-Jul-79 0830      6             442725

DUMPER tape # 1 Valid TAPE1, , Thursday, 5-Jul-79 1556
  file                                last write      size (pages)  checksum
  MISC:<HUTCHINS>DMPMSG.RNO.1          22-Jun-79 0852      5             501123
  MISC:<HUTCHINS>DUMPER.RNO.1          29-Jun-79 1559     35            737736
  MISC:<HUTCHINS>FILE1.RNO.6           3-Jul-79 1034       1             773516
  MISC:<HUTCHINS>FILE2.RNO.1           3-Jul-79 1034       1             773516
  MISC:<HUTCHINS>INV.RNO.2             2-Jul-79 1440       1             726172

End of tape.
^L
```

7. Interrupting and continuing a save operation.

When DUMPER is printing a list of all file specs as it is saving the files, interrupt the operation, specify NO FILES, and continue the save.

THE DUMPER PROGRAM

```
@DUMPER(RET)
DUMPER>TAPE (DEVICE) TAPE1:(RET)
DUMPER>REWIND(RET)
DUMPER>FILES(RET)
DUMPER>SAVE (DISK FILES)(RET)

DUMPER tape # 1  Valid TAPE1, , Friday, 15-Jun-79 1159
MISC:<HUTCHINS>
ADEXMP.TST.1 (AS) ADEXMP.TST.1 [OK]
BACKER.DOR.1 (AS) BACKER.DOR.1 [OK]
DUMP1.TST.2(CTRL/E)
Interrupting...
(AS) DUMP1.TST.2
DUMPER>NO FILES(RET)
DUMPER>CONTINUE(RET)

Total files dumped = 24
Total pages dumped = 57
DUMPER>
```

8. Checking the file contents.

Verify that all files in all savesets have transferred correctly.

```
@DUMPER(RET)
DUMPER>TAPE (DEVICE) TAPE1:(RET)
DUMPER>REWIND(RET)
DUMPER>CHECK (ALL TAPE FILES)(RET)

DUMPER tape # 1  Valid TAPE1, , Tuesday, 26-Jun-79 1159
%Compare error, page 5, file DUMP2.XMP.1
%Difference in .FBCRE of file DUMP2.XMP.1
%Difference in .FBSIZ of file DUMP2.XMP.1
%Difference in .FBWRT of file DUMP2.XMP.1
%Difference in .FBCNT of file DUMP2.XMP.1
%Difference in .FBUSW of file DUMP2.XMP.1

End of saveset
DUMPER>CHECK (ALL TAPE FILES)(RET)

DUMPER tape # 1  Valid TAPE1, , Tuesday, 26-Jun-79 1201

End of saveset
DUMPER>CHECK (ALL TAPE FILES)(RET)

DUMPER tape # 1  Valid TAPE1, , Tuesday, 26-Jun-79 1202

End of saveset
DUMPER>EXIT(RET)
@
```

THE DUMPER PROGRAM

9. Saving files from other directories to which you have access.

Do a save of all files in all accessible directories on your connected structure. Note that this example of save requires a second tape.

```
@DUMPER (RET)
DUMPER>TAPE (DEVICE) TAPE1: (RET)
DUMPER>REWIND (RET)
DUMPER>SAVE (DISK FILES) <*>*>*> (AS) <*>*>*> (RET)

DUMPER tape # 1 Valid TAPE1, , Tuesday, 12-Jun-79 0847
MISC:<BAKER.SYSERR>
MISC:<DAEMS>
MISC:<DBELL>
MISC:<DBELL.MEMO>
MISC:<DNEFF>
MISC:<DNEFF.LINK>
MISC:<DNEFF.TV>
MISC:<EIBEN>
MISC:<EKLUND>
MISC:<FECZKO>
MISC:<GATULIS>
MISC:<GILBERT>
MISC:<GILBERT.LINK>
MISC:<GILBERT.MACRO>
MISC:<GILBERT.NEW>
MISC:<GILBERT.OLD>
MISC:<GRANT>
MISC:<HEMPHILL>
MISC:<HOFFMAN.ARCHIVE>
MISC:<HOFFMAN.FORMS>
MISC:<HOUK>
MISC:<HUTCHINS>
MISC:<JCLARK>
MISC:<KECK>
MISC:<LSCD>
MISC:<LYONS>
[Mounting next tape volume]

DUMPER tape # 2 TAPE2, , Tuesday, 12-Jun-79 0901
MISC:<LYONS.DN87.700-PDP8>
MISC:<MAGRATH>
MISC:<MANUALS>
MISC:<MATSON>
MISC:<MATSON.TEXT>
MISC:<ORAN.DECNET>
MISC:<PORADA>
MISC:<R.ACE>
MISC:<REYNOLDS>
MISC:<SAUTER.NETGEN>
MISC:<SCHMITT>
MISC:<TOOLS>
MISC:<YODER>
MISC:<ZIMA.SPR>

Total files dumped = 2249
Total pages dumped = 15704
DUMPER>
```

THE DUMPER PROGRAM

10. Leaving DUMPER.

Exit from DUMPER and check the number of volumes now in the tape set originally identified as TAPE1:.

```
DUMPER>exit(RET)
@INFORMATION (ABOUT) VOLUMES (OF TAPE) TAPE1:(RET)
Volumes of tape set TAPE1: TAPE1,TAPE2
@
```

11. Recovering from a crash.

If the system crashes while you are in the middle of a large save, mount another tape and assign it a number. Then restart the save at the file that was being processed at the time of the crash.

```
@DUMPER(RET)
DUMPER>TAPE (DEVICE) TAPE1:(RET)
DUMPER>REWIND(RET)
DUMPER>FILES(RET)
DUMPER>SAVE (DISK FILES)(RET)

DUMPER tape # 2 Valid TAPE1, , Thursday, 5-Jul-79 1541
MISC:<HUTCHINS>
ADEXMP.TST.1 (AS) ADEXMP.TST.1 [OK]
DMPMSG.RNO.1 (AS) DMPMSG.RNO.1 [OK]
DUMP.XMP.2 (AS) DUMP.XMP.2 [OK]
DUMP1.TST.1 (AS) DUMP1.TST.1 [OK]
DUMP2.TST.1 (AS) DUMP2.TST.1 [OK]
DUMP3.TST.1 (AS) DUMP3.TST.1 [OK]
DUMP4.TST.1 (AS) DUMP4.TST.1 [OK]
DUMPER.RNO.1 (AS)

%IECSYSTEM-20 NOT RUNNING

System restarting, wait...
[From OPERATOR on line 211: SYSTEM IN OPERATION]
*
*
*
@MOUNT TAPE CONTAP: /WRITE-ENABLED(RET)
[Mount Request CONTAP Queued, Request-ID 147]
[Tape set CONTAP, volume CONTAP mounted]
[CONTAP: defined as MT0:]
@DUMPER(RET)
DUMPER>TAPE (DEVICE) CONTAP:(RET)
DUMPER>SSNAME CON. OF SAVE(RET)
DUMPER>REWIND(RET)
DUMPER>INITIAL (FILESPEC) DUMPER.RNO(RET)
DUMPER>SET TAPE-NUMBER (DECIMAL NUMBER) 2(RET)
DUMPER>FILES(RET)
DUMPER>SAVE (DISK FILES)(RET)
```

THE DUMPER PROGRAM

```

DUMPER tape # 2 Valid CONTAP, CON. OF SAVE, Thursday, 5-Jul-79 1603
MISC:<HUTCHINS>
DUMPER.RNO.1 (AS) DUMPER.RNO.1 [OK]
DUMPER.TST.1 (AS) DUMPER.TST.1 [OK]
EXAMP1.TST.1 (AS) EXAMP1.TST.1 [OK]
EXAMP2.TST.1 (AS) EXAMP2.TST.1 [OK]
EXDMP.XMP.1 (AS) EXDMP.XMP.1 [OK]
INV.RNO.2 (AS) INV.RNO.2 [OK]
INV.TST.1 (AS) INV.TST.1 [OK]
PHOTO.EXE.1 (AS) PHOTO.EXE.1 [OK]

Total files dumped = 15
Total pages dumped = 294
DUMPER>

```

Note that any status commands (such as FILES) that were used before the crash, must be specified again.

Examples of RESTORE:

12. Restoring a saveset saved on two volumes.

Give the MOUNT TAPE command. When the first volume is mounted, define the set as MTA-DUMPER:. Run DUMPER and restore the entire saveset to disk. Request DUMPER to list the file specs as it restores them.

```

@MOUNT TAPE SET1: /LABEL-TYPE:UNLABELED /WRITE-ENABLED /VOLIDS:TAPE1,TAPE2 (RET)
[Mount Request SET1 Queued, Request-ID 137]
[Tape set SET1, volume TAPE1 mounted]
[SET1: defined as MT0:]
@DEFINE MTA-DUMPER: SET1: (RET)
@DUMPER (RET)
DUMPER>REWIND (RET)
DUMPER>FILES (RET)
DUMPER>RESTORE (TAPE FILES) (RET)

```

```

DUMPER tape # 1 Valid TAPE1, , Monday, 2-Jul-79 0829
Loading file(s) into MISC:<HUTCHINS>
MISC:<HUTCHINS>BACKER.DOR.1;P777700;A341 (TO) BACKER.DOR.1 [OK]
MISC:<HUTCHINS>DMFMSG.RNO.1;P777700;A341 (TO) DMFMSG.RNO.1 [OK]
MISC:<HUTCHINS>DUMP.XMP.1;P777700;A341 (TO) DUMP.XMP.1 [OK]
MISC:<HUTCHINS>DUMP1.TST.2;P777700;A341 (TO) DUMP1.TST.2 [OK]
MISC:<HUTCHINS>DUMP2.TST.2;P777700;A341 (TO) DUMP2.TST.2 [OK]
MISC:<HUTCHINS>DUMP2.XMP.1;P777700;A341 (TO) DUMP2.XMP.1 [OK]
MISC:<HUTCHINS>DUMP3.TST.2;P777700;A341 (TO) DUMP3.TST.2 [OK]
MISC:<HUTCHINS>DUMP3.XMP.1;P777700;A341 (TO) DUMP3.XMP.1 [OK]
[Mounting next tape volume]

```

```

DUMPER tape # 2 Valid TAPE2, , Monday, 2-Jul-79 0843
MISC:<HUTCHINS>FILE1.RNO.6;P777700;A341 (TO) FILE1.RNO.6 [OK]
MISC:<HUTCHINS>FILE2.RNO.1;P777700;A341 (TO) FILE2.RNO.1 [OK]
MISC:<HUTCHINS>INV.RNO.2;P777700;A341 (TO) INV.RNO.2 [OK]
MISC:<HUTCHINS>PHOTO.EXE.1;P777700;A341 (TO) PHOTO.EXE.1 [OK]

End of saveset
DUMPER>

```

THE DUMPER PROGRAM

13. Restoring a single file.

Specify a single file of a saveset to be restored to disk.

```
@DUMPERRET
DUMPER>TAPE (DEVICE) SET1:RET
DUMPER>REWINDRET
DUMPER>FILESRET
DUMPER>RESTORE (TAPE FILES) FILE2.RNO.* (TO)RET

DUMPER tape# 1 Valid SET1, , Thursday, 5-Jul-79 1029
Loading file(s) into MISC:<HUTCHINS>
MISC:<HUTCHINS>FILE2.RNO.1;P777700;A341 (TO) FILE2.RNO.1 [OK]

End of saveset
DUMPER>
```

14. Restoring files saved from another structure.

Restore all files that were saved from a mountable structure (such as MISC:).

```
@DUMPERRET
DUMPER>TAPE (DEVICE) SET1:RET
DUMPER>REWINDRET
DUMPER>FILESRET
DUMPER>RESTORE (TAPE FILES) MISC:<HUTCHINS>*. * (TO)RET

DUMPER tape # 1 Valid SET1, , Thursday, 5-Jul-79 0833
Loading file(s) into MISC:<HUTCHINS>
MISC:<HUTCHINS>FILE1.TXT.1;P777700;A341 (TO) MISC:FILE1.TXT.1 [OK]
MISC:<HUTCHINS>FILE2.TXT.1;P777700;A341 (TO) MISC:FILE2.TXT.1 [OK]
MISC:<HUTCHINS>FILE3.TXT.1;P777700;A341 (TO) MISC:FILE3.TXT.1 [OK]

End of saveset
DUMPER>
```

15. Restoring multiple files.

Restore only files with file types .RNO and .DMP.

```
@DUMPERRET
DUMPER>TAPE (DEVICE) SET1:RET
DUMPER>REWINDRET
DUMPER>FILESRET
DUMPER>RESTORE (TAPE FILES) *.RNO,*.DMPRET
```


THE DUMPER PROGRAM

```
DUMPER tape # 1  Valid SET1, , Monday,  2-Jul-79 0832
Loading file(s) into MISC:<HUTCHINS>
MISC:<HUTCHINS>DMPMSG.RNO.1;P777700;A341 (TO) DMPMSG.RNO.1      [OK]
MISC:<HUTCHINS>FILE1.RNO.6;P777700;A341 (TO) FILE1.RNO.6      [OK]
MISC:<HUTCHINS>FILE2.RNO.1;P777700;A341 (TO) FILE2.RNO.1      [OK]
MISC:<HUTCHINS>INV.RNO.2;P777700;A341 (TO) INV.RNO.2          [OK]
MISC:<HUTCHINS>DUMP1.DMP.1;P777700;A341 (TO) DUMP1.DMP.1      [OK]
MISC:<HUTCHINS>DUMP2.DMP.1;P777700;A341 (TO) DUMP2.DMP.1      [OK]
MISC:<HUTCHINS>DUMP3.DMP.1;P777700;A341 (TO) DUMP3.DMP.1      [OK]

End of saveset
DUMPER>
```

16. Restoring files saved from another directory into your connected directory.

Specify that files saved from the directory PS:<ADLEY> be restored to your connected directory, MISC:<HUTCHINS>.

```
@DUMPER<RET>
DUMPER>TAPE (DEVICE) TAPE1:<RET>
DUMPER>REWIND<RET>
DUMPER>FILES<RET>
DUMPER>RESTORE (TAPE FILES) PS:<ADLEY>*,*,* (TO) MISC:<HUTCHINS>*,*,*<RET>

DUMPER tape # 1  Valid TAPE1, , Monday,  9-Jul-79 1505
Loading file(s) into MISC:<HUTCHINS>
PS:<ADLEY>COMMANDS.TV.6;P777700;A341 (TO) MISC:<HUTCHINS>COMMANDS.TV.6 [OK]
PS:<ADLEY>EMACS.INIT.2;P777700;A341 (TO) MISC:<HUTCHINS>EMACS.INIT.2 [OK]
PS:<ADLEY>LOGIN.CMD.10;P777700;A341 (TO) MISC:<HUTCHINS>LOGIN.CMD.10 [OK]
PS:<ADLEY>MAIL.TXT.1;P777700;AMONITOR (TO) MISC:<HUTCHINS>MAIL.TXT.1 [OK]

End of saveset
DUMPER>
```

17. Restoring files accessed within a period of time.

Restore only files that were accessed between 1-June-79 and 15-July-79. Also direct DUMPER to take the file account from tape.

```
@DUMPER<RET>
DUMPER>TAPE (DEVICE) TAPE1:<RET>
DUMPER>REWIND<RET>
DUMPER>FILES<RET>
DUMPER>ACCOUNT (OF RESTORED FILES FROM) TAPE<RET>
DUMPER>ASINCE (DATE AND TIME) 6-1-79 8:00AM<RET>
DUMPER>ABEFORE (DATE AND TIME) 7-15-79 5:00PM<RET>
DUMPER>RESTORE (TAPE FILES)<RET>
```


THE DUMPER PROGRAM

```

DUMPER tape # 1 Valid TAPE1, , Wednesday, 11-Jul-79 1204
Loading file(s) into MISC:<HUTCHINS>
MISC:<HUTCHINS>DMPMSG.RNO.1;P777700;A341 (TO) DMPMSG.RNO.1 [OK]
MISC:<HUTCHINS>DUMP.XMP.1;P777700;A341 (TO) DUMP.XMP.1 [OK]
MISC:<HUTCHINS>DUMPER.RNO.1;P777700;A341 (TO) DUMPER.RNO.1 [OK]
MISC:<HUTCHINS>EXDMP.EXE.4;P777700;A341 (TO) EXDMP.EXE.4 [OK]
MISC:<HUTCHINS>EXDMP.QXE.3;P777700;A341 (TO) EXDMP.QXE.3 [OK]
MISC:<HUTCHINS>INV.RNO.2;P777700;A341 (TO) INV.RNO.2 [OK]

End of saveset
DUMPER>

```

18. Skipping over existing savesets.

Rewind the tape and skip to the third saveset. Restore the files from the third saveset.

```

@DUMPER RET
DUMPER>TAPE (DEVICE) SET1: RET
DUMPER>REWIND RET
DUMPER>SKIP (NUMBER OF SAVESETS) 2 RET

DUMPER TAPE # 1 VALID SET1, , Thursday, 24-May-79 0839
DUMPER tape # 1 Valid SET1, , Thursday, 31-May-79 01427
DUMPER tape # 1 Valid SET1, , Tuesday, 5-Jun-79 0940
DUMPER>FILES RET
DUMPER>RESTORE (TAPE FILES) RET

```

```

DUMPER tape # 1 Valid SET1, , Tuesday, 5-Jun-79 0940
Loading file(s) into MISC:<HUTCHINS>
MISC:<HUTCHINS>DMPMSG.RNO.1;P777700;A341 (TO) DMPMSG.RNO.1 [OK]
MISC:<HUTCHINS>FILE1.RNO.6;P777700;A341 (TO) FILE1.RNO.6 [OK]
MISC:<HUTCHINS>FILE2.RNO.1;P777700;A341 (TO) FILE2.RNO.1 [OK]
MISC:<HUTCHINS>INV.RNO.2;P777700;A341 (TO) INV.RNO.2 [OK]

End of saveset
DUMPER>

```

19. Switching to the second volume of the tape set.

When you do not see the file you want on the first tape, rewind it and switch to the second volume. Restore the file needed, exit from DUMPER, and dismount the second volume by specifying the tape set name. Note that the DEFINE MTA-DUMPER: command is still effective with the second volume in the set.

THE DUMPER PROGRAM

NOTE

If a save has not previously been done with the FULL-INCREMENTAL switch, specifying the INCREMENTAL switch will have the same results as specifying /FULL-INCREMENTAL (i.e., DUMPER saves all files).

/NOINCREMENTAL to request that DUMPER not perform an incremental save. In this case, DUMPER saves all files, but does not preserve any information regarding the number of times the file has been saved. (/NOINCREMENTAL is the default.)

/ARCHIVE to request that DUMPER save only files that have been marked for archival.

/MIGRATE to request that DUMPER save only files that have been marked for involuntary migration by the REAPER program (refer to the TOPS-20 Operator's Guide).

NOTE

In using any of these switches, you do not have the option of renaming files from disk to tape. A file transferred to tape has the same name as the file on disk.

- You can RESTORE or RETRIEVE files from an archive or migration tape.
- You can use the CREATE command to recreate an entire directory that has been deleted.
- When you type SAVE or RESTORE with no specified directory, the default is <*> on the connected structure. (For a nonprivileged user, the default is the connected directory and structure.)
- When you restore files, the last writer string of the file on disk is set to the name of the last writer at the time of the save. (For a nonprivileged user, the last writer string is set to the name of the user doing the restore.)

Sections 7.6.1 through 7.6.5 describe the use of these commands and switches. This manual does not include step-by-step procedures for performing a system backup, restore, archival run, or migration run. For details, refer to the TOPS-20 Operator's Guide.

THE DUMPER PROGRAM

7.6.1 Backing Up System Files and/or Other Users' Files

To minimize loss of disk files, you should put backup copies of all files on magnetic tape. In many installations, this should be done on a daily basis. (for details, refer to the TOPS-20 System Manager's Guide.)

Backup is done with the SAVE command. As a privileged user you have the option of doing one of three types of saves by using SAVE plus either /FULL-INCREMENTAL, /INCREMENTAL, or /NOINCREMENTAL.

A sample backup routine for an installation might be:

Friday night: Run DUMPER to save all system files and directories using /FULL-INCREMENTAL.

All other nights: Using /INCREMENTAL:2, save all files with changes made since the save of the previous night. This allows you a margin of error in recovering files that have been inadvertently deleted and expunged.

The incremental runs can be made either more or less frequently at the discretion of the installation. In the event of a total file system loss, the installation can restore the disk to the state of the most recent incremental by first restoring the full-incremental tapes to a fresh set of packs, and then restoring any incremental tapes that were made since the full-incremental.

If the installation desires two sets of backup tapes (to gain an extra measure of safety), the above procedure can be modified as follows:

Friday night: Run a /FULL-INCREMENTAL, followed immediately by another /FULL-INCREMENTAL. This creates two sets of full-save tapes.

All other nights: Make two consecutive runs with /INCREMENTAL:2. This creates two identical sets of incremental save tapes, unless any file has been changed between saves.

If your installation has structures other than PS:, the public structure, it is important to back up files on all structures. It is advisable to use a different tape for each structure. Furthermore, it is important to use a different tape for every function: backup, migrate, and archive.

Example:

Mount two tapes or tape sets. On the first tape, save all files in all directories on the public structure, PS:. On the second tape, save all files in all directories on the private structure, MISC:.

```
@ENABLE RET
$MOUNT TAPE SAVE1: /LABEL-TYPE:UNLABELED /WRITE-ENABLED RET
[Mount Request SAVE1 Queued, Request-ID 283]
[Tape set SAVE1, volume SAVE1 mounted]
[SAVE1: defined as MT0:]
$DUMPER RET
DUMPER>TAPE (DEVICE) SAVE1: RET
DUMPER>SSNAME PS:-FULL SAVE RET
DUMPER>REWIND RET
DUMPER>SAVE (DISK FILES) /FULL-INCREMENTAL PS:<*>*.*. * RET
```

THE DUMPER PROGRAM

```

DUMPER tape # 1 Valid SAVE1, PS:-FULL SAVE, Wednesday, 11-Jul-79 1313
PS:<1EIBEN>
PS:<1STUDENT>
PS:<ABEL>
PS:<ACARLSON>
PS:<ACCOUNTS>
PS:<ACE>
PS:<ACT10>
PS:<ADLEY>
PS:<ALLEN>
PS:<ANDERSON>
PS:<APLB>
PS:<APLSF>
PS:<ARNOLD>
.
.
.

DUMPER>EXIT RET
$ MOUNT TAPE SAVE2: /LABEL-TYPE:UNLABELED /WRITE-ENABLED RET
[Mount Request SAVE2 Queued, Request-ID 287]
[Tape set SAVE2, volume SAVE2 mounted]
[SAVE2: defined as MT0:]
$ DUMPER RET
DUMPER>TAPE (DEVICE) SAVE2: RET
DUMPER>SSNAME MISC:-FULL SAVE RET
DUMPER>REWIND RET
DUMPER>SAVE (DISK FILES) /FULL-INCREMENTAL MISC:<*>*.**.* RET

DUMPER tape # 1 Valid SAVE2, MISC:-FULL SAVE, Wednesday, 11-Jul-79 1324
MISC:<1BELANGER>
MISC:<1EIBEN>
MISC:<2BELANGER>
MISC:<2EIBEN>
MISC:<ACE>
MISC:<ADLEY>
MISC:<APLB>
MISC:<APLSF>
MISC:<BAKER>
MISC:<BAKER.SYSERR>
MISC:<BATOR>
MISC:<BELANGER>
.
.
.
DUMPER>EXIT RET
$

```

If the system crashes while you are saving, restoring, archiving, or migrating files, you do not have to start from the beginning when the system is restarted. As for a nonprivileged user, enter DUMPER and type the DUMPER commands INITIAL and SET TAPE-NUMBER. (Refer to Example 11 in Section 7.5.5.)

THE DUMPER PROGRAM

7.6.2 Restoring Files and Directories from System Backup Tapes

If a user accidentally deletes files from the disk area, he will want to restore them from backup tapes. In many installations, you as a privileged user or operator must restore these files.

To fill his request, mount the full save tape (of files saved on a weekly basis) and restore the requested files. Then, if necessary, mount the incremental save tape (of new or changed files saved on a daily basis) and restore the remainder of files in the user's request. The RESTORE command is the same as for a nonprivileged user. You must specify the source in order to transfer the files of the user who made the request.

Example:

```
@DUMPER RET
DUMPER>TAPE (DEVICE) TAPE1: RET
DUMPER>REWIND RET
DUMPER>FILES RET
DUMPER>RESTORE (TAPE FILES) MISC:<HUTCHINS>*.RNO.* (TO) MISC:<HUTCHINS> RET

DUMPER tape # 1 Valid TAPE1, , Tuesday, 5-Jun-79 0940
Loading file(s) into MISC:<HUTCHINS>
MISC:<HUTCHINS>DMPMSG.RNO.1:P777700:A341 (TO) MISC:<HUTCHINS>DMPMSG.RNO.1 [OK]
MISC:<HUTCHINS>DUMPER.RNO.1:P777700:A341 (TO) MISC:<HUTCHINS>DUMPER.RNO.1 [OK]
MISC:<HUTCHINS>FILE1.RNO.6:P777700:A341 (TO) MISC:<HUTCHINS>FILE1.RNO.6 [OK]
MISC:<HUTCHINS>FILE2.RNO.1:P777700:A341 (TO) MISC:<HUTCHINS>FILE2.RNO.1 [OK]
MISC:<HUTCHINS>INV.RNO.2:P777700:A341 (TO) MISC:<HUTCHINS>INV.RNO.2 [OK]

End of. saveset
DUMPER>
```

If a user's entire directory is accidentally deleted, you can restore it with the DUMPER command CREATE. Having entered DUMPER, type CREATE before RESTORE. As DUMPER restores all files and directories, it restores (creates) the deleted directory exactly as the directory was saved. (When you do a save with privileges enabled, all directory information is automatically saved.)

7.6.3 Archiving Marked Files

If your installation is using the archive/virtual disk system for off-line storage of files, the installation establishes a schedule under which it runs the DUMPER program to copy files marked for archiving onto tape. User requests to archive files do not appear at operator command level. However, you can use the TOPS-20 command INFORMATION ARCHIVE-STATUS DSK*:<*>*.*. to display the files that users have requested to be archived.

NOTE

DSK*: refers to all mounted structures currently accessible to your job.

THE DUMPER PROGRAM

```
$ INFORMATION (ABOUT) ARCHIVE-STATUS (OF FILES) DSK*:<*>*.*.*(RET)
<ALLEN>CAD.CMD.1 Migration Prohibited
<ALLEN>COMAND.CMD.1 Migration Prohibited
<BERENSON>X..1 Archived
<BLOUNT>FILE-TO-ARCHIVE.TXT.1 Archived
<DBELL>INVISIBLE-LOGIN.CMD.1 Migration Prohibited
<DBELL>MAIL.TXT.1 Migration Prohibited
<ENGEL>A.A.1 Migration Prohibited
<ENGEL>BATCH.CMD.1 Migration Prohibited
<GLXLIB>FOO.LOG.1 Archived
<GLXLIB>FOO.MAC.1 Archived
<HELLIWELL>COMAND.CMD.3 Migration Prohibited
<HESS.DOC>USAGE.TXT.1 Migration Prohibited
<HESS.DOC>USGREC.FRM.1 Migration Prohibited
<JABLONSKY>TEST.EXE.1 Archived
<KIRSCHEN>FOO.OFFLINE.1 Archived
<LAUBER>MAIL.TXT.1 Migration Prohibited
<LCAMPBELL>CASE.1.5 Archived
<LCAMPBELL>DECNET>OLD-NMAIL.REL.1 Archive requested
<LYONS>SELF.TXT.1 Migration Prohibited
<LYONS>TEAMS.WRD.1 Migration Prohibited
<MCLEAN>ATOG.WDS.1 Migration Prohibited
<OPERATOR>SYSERR.LOG.1 Archived
<OSMAN.SOURCES>J3NEW.RNO.1 Archive requested
<R.ACE>P.FIL.EXE.2 Archive requested
<REILLY>ABBREV.DEFNS.1 Migration Prohibited
<SALAS>110678.FRI.1 Archived
<SALAS>SYMSUB.MAC.1 Archived
<SNYDER, BEIGEBROOK>CRISS.RNO.1 Archived
MISC:<BROWN>BATCH1.CTL.1 Archived
MISC:<BROWN>BATCH2.CTL.1 Archived
MISC:<GILBERT>DDT.SORTED-SYMBOLS-MAC.1 Archived
MISC:<GILBERT>OLD-RELEASE-4-FILDDT.MAC.1 Archived
MISC:<HUTCHINS>ADEXMP.TST.1 Archived
MISC:<HUTCHINS>DUMP1.DMP.1 Archive requested
MISC:<HUTCHINS>DUMP1.TST.1 Archived
MISC:<HUTCHINS>DUMP2.DMP.1 Archive requested
MISC:<HUTCHINS>DUMP2.TST.1 Archived
MISC:<LATTA>MOOBE.TXT.1 Archived
MISC:<LSCD>MAIL.OLD.1 Archived
MISC:<LYONS>DDT.DIRECTORY.1 Migration Prohibited
MISC:<LYONS>DN87.DIRECTORY.1 Migration Prohibited
$
```

You then mount a tape used only for archiving and run DUMPER.

```
DUMPER> SAVE (DISK FILES)/ARCHIVE STR:<*>*.*.*(RET)
```

If no file spec is given with the command, all files marked for archiving on the connected structure are transferred to tape.

At the beginning of the archive run, DUMPER asks:

Is this a new tape?

If you answer YES, DUMPER asks:

Are you sure?

THE DUMPER PROGRAM

If you answer YES again, DUMPER writes at the beginning of the tape, erasing anything that is currently written on it. If you answer that this is not a new tape, DUMPER positions the tape after the last saveset and appends new files to the existing files.

When the marked files have been transferred, or the end-of-tape is reached, DUMPER indicates that the operation is complete and notifies you that Pass 2 has begun. Pass 2 is a check to determine if DUMPER should delete the file contents, and to update the archive status. You must mount a second tape and type another SAVE/ARCHIVE command. The purpose of the second tape is to write the files again, verify them as on the first tape, and then delete the contents of the files from disk, unless the user has requested that the contents be retained. The files are then invisible in the user's directory.

Example:

```
@DUMPER RET
DUMPER>TAPE (DEVICE) ARCH1: RET
DUMPER>REWIND RET
DUMPER>FILES RET
DUMPER>SAVE (DISK FILES) /ARCHIVE MISC:<HUTCHINS>*. *.* (AS) RET
$Is this a new tape? Y RET
$Are you sure? Y RET

DUMPER tape Valid ARCH1, Save set# 2, Tuesday, 3-Jul-79 0908
  Pass 1 started
MISC:<HUTCHINS>
DUMP1.DMP.1 (AS) DUMP1.DMP.1 [OK]
DUMP2.DMP.1 (AS) DUMP2.DMP.1 [OK]
DUMP3.DMP.1 (AS) DUMP3.DMP.1 [OK]

  Pass 1 completed.
  Saveset complete. Tape ready for storage.
  Pass 2 started.
  Pass 2 completed.

Total files dumped = 3
Total pages dumped = 3
DUMPER>EXIT RET
$DISMOUNT TAPE ARCH1: RET
[Tape dismounted, logical name arch1: deleted]
$MOUNT TAPE ARCH2: /WRITE-ENABLED RET
[Mount Request ARCH2 Queued, Request-ID 140]
[Tape set ARCH2, Volume ARCH2 mounted]
[ARCH2: defined as MT1:]
DUMPER RET
DUMPER>TAPE (DEVICE) ARCH2: RET
DUMPER>REWIND RET
DUMPER>FILES RET
DUMPER>SAVE (DISK FILES) /ARCHIVE MISC:<HUTCHINS>*. *.* (AS) RET
$Is this a new tape? Y RET
$Are you sure? Y RET
```

THE DUMPER PROGRAM

DUMPER tape Valid ARCH2, Save set# 3, , Tuesday, 3-Jul-79 0919

Pass 1 started
MISC:<HUTCHINS>
DUMP1.DMP.1 (AS) DUMP1.DMP.1 [OK]
DUMP2.DMP.1 (AS) DUMP2.DMP.1 [OK]
DUMP3.DMP.1 (AS) DUMP3.DMP.1 [OK]

Pass 1 completed.
Saveset complete. Tape ready for storage.
Pass 2 started.
Pass 2 completed.

Total files dumped = 3
Total passes dumped = 3
DUMPER> exit(RET)

WARNING

When DUMPER archives or migrates files,
it numbers the two tapes as Save set #2
and Save set #3.

7.6.4 Migrating Files

According to the procedures of your installation, you can periodically migrate files (i.e., copy them onto tape and delete them from disk). This is not done at the user's request. It is a means of clearing disk space of files that have not been referenced within a specified period of time. It is also a technique for returning directory disk usage to within its quota.

You must specify a time period with the MIGRATE command to the REAPER program. When REAPER is run, it marks, for involuntary migration, all files that have not been used during that time period.

Having run REAPER, you then run DUMPER and use the command SAVE (DISK FILES) /MIGRATE. This dumps all files marked for migration onto the mounted tape.

As with archiving, you use two tapes when migrating files. This provides a backup system in case one tape is bad or misplaced. DUMPER then follows the same routine as with archiving. It first asks if this is a new tape. When the files have been successfully migrated to the second tape, their contents are deleted from disk and DUMPER sends mail to notify him that his files have been migrated. The files then have an ;OFFLINE status in the user's directory.

THE DUMPER PROGRAM

7.6.5 Retrieving or Restoring Archived and Migrated Files

Once a user requests that the file be archived or migrated, he cannot access it because its contents have been deleted from disk (unless he has specified the subcommand RETAIN with the archive request). Only the FDB remains. The user can, however, request a retrieval of a file by using the TOPS-20 command RETRIEVE. This creates an entry in the system retrieval queue. (The queue can be displayed by typing the INFORMATION RETRIEVAL-REQUESTS command.) All retrieval requests are kept in order according to the archived or migrated tape information. This information consists of the volume identification, the tape saveset number, and the tape file number. When you are ready to process the retrieval request queue, run DUMPER and use DUMPER's RETRIEVE command:

```
DUMPER>RETRIEVE (FILES) file spec RET
```

If the file spec is omitted, DUMPER processes all requests in the queue. If a file spec is included, only those files that match the file spec are processed. (The file spec may include wildcards.) For example:

```
RETRIEVE (FILES) MISC:<HUTCHINS>*.TST RET
```

When DUMPER selects a file or files for retrieval, it submits a mount request for the tape containing the file(s). After you or the operator have mounted that tape, DUMPER begins the retrieval process. When the files have been retrieved, DUMPER notifies the requestor.

NOTE

When processing retrieval requests, you do not need to type the TOPS-20 command MOUNT TAPE or the DUMPER command TAPE. DUMPER automatically requests the tapes it needs.

If you cannot find the appropriate tape, type the OPR command CANCEL MOUNT-REQUEST. The system requeues the request using the information from the second tape. If the file cannot be recovered from either tape, the request is deleted and the requestor is notified.

Assuming all files can be found, only one RETRIEVE command is necessary to process the retrieval queue. If the files to be retrieved are on different tapes, DUMPER automatically unloads the mounted tape and submits a mount request for the next tape it needs.

WARNING

If you type RETRIEVE when there are no retrieval requests in the queue, DUMPER waits indefinitely. Whenever possible, check the queue (with the TOPS-20 command INFORMATION RETRIEVAL-REQUESTS) before typing RETRIEVE.

THE DUMPER PROGRAM

If a user deletes the filename from his directory, and therefore the File Descriptor Block (FDB), and then wishes to retrieve the file, you must use the RESTORE command. When you locate the file, you can restore it with or without the tape information. This information indicates whether the file is archived or migrated, and if so, onto which tapes. If you do not wish to restore this information with the file, type the command RESTORE/NOTAPE-INFORMATION. To reinstate the transfer of this information with the file, type the command RESTORE/TAPE-INFORMATION (this is the default).

7.7 DUMPER COMMANDS

This section contains an alphabetical list of all DUMPER commands. Each listed command includes a brief description, command type (Status-Setting, Action, or Tape-Positioning), and description of optional arguments. Where applicable, there is an indication that the command is for privileged users only.

ABEFORE (DATE AND TIME) date time Status-Setting

Saves or restores only files that were typed, printed, or read (as maintained by .FBREF) before the specified date and time.
Default: Date - none; Time - 00:00:01

ACCOUNT (OF RESTORED FILES FROM) account Status-Setting

Restores files with either the system (SYSTEM-DEFAULT) account or the account stored with the file (TAPE).
Default: TAPE

ASINCE (DATE AND TIME) date time Status-Setting

Saves or restores only files that were typed, printed, or read (as maintained by .FBREF) since the specified date and time.
Default: Date - none; Time - 00:00:01

BEFORE (DATE AND TIME) date time Status-Setting

Saves or restores only files that were created or modified (as maintained by .FBCRV and .FBWRT) before the specified date and time.
Default: Date - none; Time - 00:00:01

CHECK (ALL TAPE FILES) Action

Checks every file in the current saveset against the contents and file descriptor blocks (FDB's) of the corresponding files on disk. Corresponding files must have the same name, structure, directory, type, and generation number for the check to be made. (Refer to Table 7-3 for a list of possible differences.)

[NO] CHECKSUM (FILES) type Status-Setting

Activates or suppresses checksumming. Two types may be specified: SEQUENTIAL (for INTERCHANGE mode) or BY-PAGES (that checks every word of every page).
Default: NO CHECKSUM

THE DUMPER PROGRAM

| | |
|---|------------------|
| CONTINUE | Action |
| Continues a CHECK, RESTORE, RETRIEVE, SAVE, or any Tape-Positioning command, after you have interrupted the command with CTRL/E. | |
| [NO] CREATE (DIRECTORIES FROM TAPE DATA) | Status-Setting |
| Creates or does not create user directories from directory information on the tape. Default: NO CREATE. (Privileged User Only) | |
| CTRL/E | Action |
| Halts the action of a CHECK, RESTORE, RETRIEVE, SAVE, or tape-positioning command. DUMPER responds with INTERRUPTING... and its prompt. If you type anything other than a status-setting command, DUMPER asks if you want to abort the interrupted command. | |
| DENSITY (OF MAGTAPE) n | Status-Setting |
| Sets the tape density to the given number of bits per inch (bits/in): 200, 556, 800, 1600, 6250, or JOB-DEFAULT (set by the system command SET TAPE DENSITY). Default: The density listed in the TOPS-20 command INFORMATION (ABOUT) TAPE-PARAMETERS. | |
| [NO] DIRECTORIES | Status-Setting |
| Reactivates or suppresses printing, on your terminal, directory names as DUMPER saves or restores each directory. Default: DIRECTORIES | |
| EOT | Tape-positioning |
| Positions the mounted tape at the end of the last saveset written on the tape. DUMPER prints all existing saveset names and the message: TAPE POSITIONED AFTER LAST SAVESET | |
| EXIT | Action |
| Exits to TOPS-20 command level. (Refer to the QUIT command.) | |
| [NO] FILES | Status-Setting |
| Reactivates or suppresses printing file specs on your terminal, as DUMPER saves or restores each file. Default: NO FILES | |

THE DUMPER PROGRAM

FORMAT (VERSION NUMBER IS) n Status-Setting
Allows DUMPER to read or write tapes written with previous versions of DUMPER.
Default: Version 4

HELP Action
Prints a list of all valid DUMPER commands on your terminal.

[NO] INDUSTRY (COMPATIBLE 36-BIT MODE) Status-Setting
Allows or does not allow DUMPER to read and write tapes using the Industry-Compatible tape data mode.
Default: CORE-DUMP MODE

INITIAL (FILESPEC) file spec Status-Setting
Begins a CHECK, SAVE, or RESTORE with the specified file.

[NO] INTERCHANGE (FORMAT) Status-Setting
Allows or does not allow DUMPER to read tapes written with the TOPS-10 BACKUP program or to write tapes to be read by the TOPS-10 BACKUP program. (INTERCHANGE should never be used when writing tapes to be read by another TOPS-20 system.)
Default: DUMPER format

[NO] LIST (LOG INFORMATION ON FILE) file spec Status-Setting
Prints or does not print a list, in the specified file, of all files as DUMPER saves them.
Default: NO LIST
Default file spec: LPT:DUMPER.LOG

MBEFORE (DATE AND TIME) date time Status-Setting
Transfers only files modified (changed, created, appended, or renamed as maintained by .FBCRE) before the specified date and time.
Default: Time=00:00:01

MSINCE (DATE AND TIME) date time Status-Setting
Transfers only files modified (as maintained by .FBCRE) since the specified date and time.
Default: Time=00:00:01

PARITY (OF MAGNETIC TAPE) parity Status-Setting
Sets the parity of the mounted tape to EVEN or ODD.
Default: The parity listed in the TOPS-20 command INFORMATION (ABOUT) TAPE-PARAMETERS

THE DUMPER PROGRAM

PRINT (DIRECTORY OF TAPE ONTO FILE) destination Action

Records a printed list of filenames into the specified file. The list contains file specs of all existing files on the entire tape beginning at the current saveset.
Default: Prints on your terminal

PROTECTION (OF RESTORED FILES FROM) argument Status-Setting

Restores files with SYSTEM-DEFAULT protection or protection taken from TAPE.
Default: TAPE (if the PROTECTION command is omitted)
Argument Default: None; you must specify one or the other.

QUIT Action

Exits to TOPS-20 command level. (Refer to the EXIT command.)

RESTORE (TAPE FILES) source (TO) destination Action

Restores the magnetic tape source file(s) to disk with the given destination file specs. If you are restoring files from a directory other than your connected directory, you must specify that directory in the source file specification. If files exist in the destination directory with the same names and types as the files on tape, DUMPER restores the files according to the specification of the SUPERSEDE command: ALWAYS, NEVER, OLDER
Default: Source file spec = all files in the current saveset that were saved under the connected structure and directory; if privileged, all files in all directories on your connected structure.
Destination file spec = same names and types as the files on tape.

(Privileged User Only):

The FDB of each file contains information that reflects whether a file is archived or migrated, and, if so, the volume identifiers of the DUMPER tapes that contain the file. The /TAPE-INFORMATION switch instructs DUMPER to restore this information to the FDBs of the files that are being restored. /NOTAPE-INFORMATION requests that this information not be restored.
Default: /TAPE-INFORMATION

RETRIEVE (FILES) file spec Action

Instructs DUMPER to process the requests in the system's file retrieval queue.
Default: Process all files in the queue
(Privileged User Only)

REWIND argument Tape-Positioning

Positions the specified tape at the beginning of the tape. If the argument is CURRENT-VOLUME, DUMPER rewinds the currently mounted tape to the beginning. If the argument is SWITCHING (TO VOLUME NUMBER) n, DUMPER releases the currently mounted tape, requests volume n of the multi-reel tape set, and rewinds it to the beginning.
Default: CURRENT-VOLUME

THE DUMPER PROGRAM

SAVE (DISK FILES) source (AS) destination Action

Saves the disk source file(s) onto magnetic tape. If all files cannot fit on one tape, DUMPER requests that the next volume in the set be mounted.
Default: Source file spec = all files in your connected directory; if privileged, all files in all directories on your connected structure.
Destination file spec = same names and types as the files on disk.

(Privileged User Only):

The following switches can be appended to the SAVE command:

- /FULL-INCREMENTAL for a system backup of all files. (This also marks all files as having been saved.)
- /INCREMENTAL:n for a daily backup of files that either have not been saved on at least n tapes, or have been modified since the last INCREMENTAL or FULL-INCREMENTAL run.
Default: /INCREMENTAL:1
- /NOINCREMENTAL for a backup of all files.
- /ARCHIVE for archival of marked files only.
- /MIGRATE for involuntary migration of marked files only.

Default: /NOINCREMENTAL

SET BLOCKING-FACTOR (TO) n (RECORDS) Status-Setting

Sets the number of logical records per physical record written on tape by DUMPER. The range must be between 1 and 15.
Default: 1 record

SET TAPE-NUMBER (DECIMAL NUMBER) n Status-Setting

Assigns a number to a new tape when you continue a restore or save after a crash, or when you restore a file nonsequentially from a multi-reel saveset.

[NO] SILENCE Status-Setting

Activates or suppresses the printing of directory names and file specs on your terminal, as files are saved or restored. NO SILENCE is equivalent to FILES and DIRECTORIES.
Default: NO FILES, DIRECTORIES

SINCE (DATE AND TIME) date time Status-Setting

Transfers only files created or whose contents were changed since the specified date and time.
Default: Time = 00:00:01

THE DUMPER PROGRAM

SKIP (NUMBER OF SAVESETS) n Tape-Positioning

Skips over n savesets. The skip can be directed forward (n), backwards (-n), or to the beginning of the current saveset (0).

SSNAME name Status-Setting

Specifies the name to be written in the saveset header on the tape. The name can be any string of up to 30 characters. It is printed on your terminal whenever you save or restore files.
Default: Existing saveset name, if any.

SUPERSEDE condition Status-Setting

Sets the condition under which DUMPER rewrites a disk file with a magnetic tape file of the same name and file type. You must specify one of three conditions:

1. ALWAYS - to supersede the file on disk regardless of the last write date or generation number of that file.
2. NEVER - to never supersede the disk file; the specified file is written to disk only if there is no existing disk file with the same name and file type.
3. OLDER - to supersede the disk file only when the file on tape is newer (i.e., has a later write date and/or a higher generation number than the file on disk).

Default: SUPERSEDE OLDER (if the SUPERSEDE command is omitted)

TAPE (DEVICE) name: Status-Setting

Specifies the tape device to be used for file transfers. The name can be either a physical device (MTl:) or a logical name (TAPEl:).

NOTE

If you define the mounted tape as MTA-DUMPER:, you can omit the DUMPER command TAPE.

UNLOAD

Used to rewind a tape onto the source reel if your installation does not have tape drive allocation enabled.

THE DUMPER PROGRAM

7.8 DUMPER MESSAGES

This section contains an alphabetical listing of all messages generated by DUMPER. Included with each message are a description and a suggested user response. There is also identification of messages that appear to privileged users only. The messages printed by DUMPER fall into two general categories: warning messages and fatal errors. Warning messages are preceded by a percent sign (%) and indicate that something unexpected occurred, but that DUMPER was able to recover. In this case, verify that the process in progress at the time of the warning is correct. Fatal errors are preceded by a question mark (?), and indicate an occurrence that DUMPER could not handle. In this case, DUMPER aborts the operation. You must fix the problem before reissuing your command string. The % and ? preceding the messages are easily detected in Batch jobs.

Some of the messages contain information that is dependent on the exact command string or file you specified. These message variables are as follows:

| | |
|------------|---|
| [device] | A device name. |
| [<dir>] | The name of a directory that could not be created. |
| [file] | A file spec. |
| [location] | The FDB entry that is different in a tape or disk file from the corresponding file on disk or tape. |
| [n] | The number of a page or record. |
| [reason] | The reason for a file access failure. |
| [result] | The result of a magtape data error. |
| [void] | The identification number of a tape reel. |

?Ambiguous

Description: You typed a command that requires an argument (such as CHECKSUM) but omitted the argument.

Suggested User Response: Retype the command including the appropriate argument. If you do not know what the choices for the argument are, type the command followed by a ?.

%ARCF failure, Invalid function

Description: (Privileged User Only) You tried to use the RESTORE command with an archived file, or for some other reason, DUMPER is not able to transfer the contents of the FDB to the disk file.

Suggested User Response: Reexecute the operation using the RETRIEVE command.

?ATTEMPT TO CREATE DIRECTORIES FROM A TENEX TAPE

Description: You specified the CREATE command for a restore, but the tape is in an old format.

Suggested User Response: Build any directories that you need by hand. Do not specify CREATE for the restore.

THE DUMPER PROGRAM

?BAD RESTART FILE. MARK THIS TAPE AS FULL AND USE NEW TAPE FOR FURTHER ARCHIVALS.

Description: (Privileged User Only) You used a restart file during archivals that could not be opened.

Suggested User Response: Build any directories that you need by hand. Do not specify CREATE for the restore.

?BAD VOLUME IDENTIFIER

Description: You typed an unidentified valid with /VOLIDS:.

Suggested User Response: Retype or ensure that your valid will be acknowledged by the operator.

%BEGINNING OF TAPE ENCOUNTERED

Description: During a SKIP-n command, DUMPER reached the beginning of the tape before it skipped the specified number of savesets.

?BLOCKING-FACTOR too high for current tape density

%Tape will be written with BLOCKING-FACTOR of [n]

Description: Records become too long if you use the specified blocking factor. DUMPER sets the BLOCKING-FACTOR to a decimal number that the tape can accomodate (between 1 and 15).

Suggested User Response: Specify a smaller blocking factor.

?Blocking factor must be greater than 0

Description: You specified a blocking factor that was not 1 or greater.

Suggested User Response: Reissue the command specifying a positive blocking factor.

?Cannot backspace on labeled tape

Description: You attempted to skip backwards on a labeled tape.

Suggested User Response: Do a REWIND SWITCHING 1, then skip forward to the desired saveset.

?CANNOT CLOSE LOG FILE [file]

Description: The close of the specified log file failed at the end of a save operation. DUMPER stopped logging to the file.

Suggested User Response: First, exit from DUMPER. Then, give the INFORMATION FILE-STATUS command to find the file and give the CLOSE command. Then, CONTINUE and give the LIST command again.

?CANNOT CREATE RESTART FILE

Description: DUMPER could not open the restart file during an archiving run.

Suggested User Response: Try the operation again. If the error persists, contact your Software Specialist or send a Software Performance Report (SPR) to DIGITAL.

THE DUMPER PROGRAM

%Cannot find all the pages of file [file], not dumped because:

Description: The number of pages found in the specified file is less than the count indicated by the FDB.

Suggested User Response: None. To prevent this in the future, do an EXPUNGE of the directory with a PURGE subcommand to recompute the count in the FDB.

?CANNOT CONTINUE

Description: You typed CTRL/E to halt the execution of RESTORE, CHECK, RETRIEVE, SAVE, or any Tape-Positioning command. You then typed a command other than DIRECTORIES, FILES, HELP, or SILENCE (preventing DUMPER from being able to continue with the halted operation) followed by CONTINUE.

Suggested User Response: Rewind the tape with the REWIND command and repeat the original command.

%Cannot get JFN for file [file] because:
[reason]

Description: You do not have access to the specified file for a transfer, check, etc., because there is not enough space on the disk to open the file.

Suggested User Response: First determine whether your own directory is full, or other users' directories on the system are full. To do this, type the INFORMATION (ABOUT) DISK-USAGE (OF DIRECTORY) dev:<directory>. If your own directory is full, you must delete and expunge files. If other users' directories are full, you must wait until space is cleared.

%Cannot load file [file] because:
[reason]

Description: DUMPER could not load the specified file from tape onto the disk area. The reason is specified.

Suggested User Response: Fix the reason specified for the error and try the operation again.

?Cannot open file [file] because:
[reason]

Description: There is an OPENF failure for the file. The reason is specified.

Suggested User Response: If you have wheel or operator capabilities, enable them and try the operation again. If you do not have these capabilities, omit the operation.

?Cannot read labeled tapes in BYPASS mode

Description: You attempted to read a labeled tape while bypassing tape labeling.

Suggested User Response: Remount the tape, but do not specify /LABEL:BYPASS in the MOUNT command.

THE DUMPER PROGRAM

?Cannot suppress tape information for a retrieval

Description: (Privileged User Only) You attempted to retrieve files from ARCHIVE or MIGRATION tapes without their tape information.

Suggested User Response: Do not use /NOTAPE-INFORMATION in the RESTORE command.

?Cannot switch to next tape volume because:
[reason]

Description: Your attempt to switch volumes for a labeled tape failed.

Suggested User Response: Clear up the problem specified in [reason] and try again.

?Carriage return required.

Description: You typed a comma, indicating more than one file, and then ended the string with a nonalphanumeric character other than a carriage return.

Suggested User Response: Retype the string without the comma and extraneous file specs, if any. Terminate the string by pressing RETURN.

?Checksum error, record [n]

Description: The checksum is wrong for the specified record number although there is no data error.

Suggested User Response: Verify that the files you are trying to save or restore are correct. If not, save them again or restore them from another tape.

%Compare error, page [n], file [file]

Description: There is a file in which the data on tape differs from the data on disk. This message is printed only if the CHECK command has been typed. The n indicates the page number of the file on disk, and file spec is the file in which the error occurs.

Suggested User Response: Try to transfer the file again. Note, however, that this error is often caused by a program that updated the file after you saved it on tape.

?Device error, possible incorrect density.
Type <CR> to try again.

Description: You typed the PRINT, RESTORE, SKIP, EOT, or CHECK command to DUMPER when:

- There was nothing written on tape,
- You were using INTERCHANGE when the tape was not interchange, or vice versa.
- The tape density was incorrect, or
- The tape was on a faulty drive.

THE DUMPER PROGRAM

Suggested User Response: Type CTRL/E and try another command, exit from DUMPER and mount another tape, or check and respecify the density.

?DEVICE MUST BE DISK

Description: You attempted to save files from a non-disk device.

Suggested User Response: Use only disk devices.

%Difference in [location] of file [file]

Description: There is a difference in the indicated FDB location between the file on disk and the file on tape. The FDB contains information describing the history and structure of the file. Refer to Table 7-3 for common locations and the meaning when there is a difference.

Suggested User Response: Try to transfer the file again. If the error still occurs, contact the operator or system manager.

%Directory [<dir>] not created because, [reason]

Description: A directory could not be created to hold the files to be transferred. The reason is specified.

Suggested User Response: Create the directory with the BUILD command and try again. Or, restore the files to a different, existing directory.

?DISK ERROR IN FILE [file], PAGE [n]

Description: A disk file data error occurred on the specified page. The file is bad.

Suggested User Response: To stop this error, correct the bad file by copying it over itself or re-creating it.

%DISK ERROR IN FILE [file], PAGE TABLE

Description: A data error occurred in the index block for the file. The file is bad.

Suggested User Response: To stop this error, correct the bad file by copying it over itself or re-creating it.

?Disk full or quota exceeded, Type <CR> to attempt to continue

Description: There is no disk space available in your area to restore the file(s).

Suggested User Response: If you are still unable to continue after typing RETURN, type CTRL/C to exit to TOPS-20 command level. Delete and expunge all unnecessary files. Then type CONTINUE to return to DUMPER, and try typing RETURN again.

THE DUMPER PROGRAM

%Do you really want to abort your interrupted command?

Description: You typed CTRL/E, followed by a DUMPER command other than a Status-Setting command.

Suggested User Response: You must respond with either Y (yes) or N (no). If you type any other response, DUMPER prints ?YES or NO only. If you type N, you can then type CONTINUE and the interrupted command continues. If you type Y, the current command process is halted, and you can enter a new command in response to the DUMPER prompt.

?Does not match switch or keyword

Description: You typed an unrecognized argument with a command that takes either an optional or mandatory argument (for example, ACCOUNT).

Suggested User Response: Retype the command, including an appropriate argument. If you do not know which arguments are acceptable for this command, type the command, followed by a space and a question mark.

?Drive probably off-line, type <CR> when ready to try again.

Description: One of two problems exists:

1. The operator has not placed the tape drive on line.
2. You are attempting to read the tape at the wrong density.

Suggested User Response: Make sure the operator has the tape on line. Or type a CTRL/C, restart DUMPER, and reset the density with the DENSITY (OF MAGTAPE) command.

NOTE

If you cannot recall the correct density, try each one: 200-BPI, 556-BPI, 800-BPI, 1600-BPI, and 6250-BPI.

%Duplicate record encountered, record [n] ignored.

Description: Each record on tape has a sequence number. When DUMPER reads a record, it checks to see if the sequence number is one greater than the previous record. DUMPER prints the above message if the sequence number is the same as the previous record. The sequence number is the same, because a record error was encountered when the tape was originally written. For each error encountered, DUMPER created a duplicate record (refer to the message %WRITE ERROR ON TAPE, RECORD n, WRITING DUPLICATE RECORD). When you restore the files from that tape, DUMPER reads the bad record first. Since normal reading operates at a greater fault tolerance than write-checking, DUMPER may be able to read a

THE DUMPER PROGRAM

record that was previously thought to be bad. If this occurs, DUMPER takes the first record and ignores the duplicate (second) record. If the first record is bad, DUMPER prints either ?MAGTAPE DATA ERROR, ?MAGTAPE UNKNOWN ERROR, ?MAGTAPE CHECKSUM ERROR, or another similar message. DUMPER then reads the next record. If the duplicate record is correct, DUMPER prints the message RECOVERED. If the error is not recovered, DUMPER prints a message indicating the location of the bad record and continues the transfer.

Suggested User Response: Ignore this warning message.

%End of tape encountered

Description: You typed EOT or SKIP n, and the tape advanced to the end-of-tape marker.

Suggested User Response: Type REWIND or SKIP -n to return to the beginning of the tape, or to a previous saveset.

?End of tape, mount next reel

Description: (Privileged User Only) You accessed a tapeset and the currently-mounted reel is finished. DUMPER prompts you to mount the next volume of the set.

Suggested User Response: Mount the next volume.

%End of tape reached and requested file not found

Description: You attempted to locate a file that does not exist on the currently-mounted tape.

Suggested User Response: Try another tape on which the file might have been saved.

?ERROR POSITIONING TAPE:

Description: Your backspace operation failed on a labeled tape.

Suggested User Response: Try the operation again. If it still fails, contact your Software Specialist or send a Software Performance Report (SPR) to DIGITAL.

?Error rewinding tape, [reason]

Description: DUMPER failed to rewind the current tape for the specified reason.

Suggested User Response: Fix the reason for the failure and try again.

?Error switching volumes

Description: A volume switch failed on a labeled tape.

Suggested User Response: Fix the reason for the failure and try again.

THE DUMPER PROGRAM

%FAILED IN ATTEMPT TO DELETE RESTART FILE

Description: DUMPER could not find the restart file while trying to delete it during an archive run.

Suggested User Response: Exit from DUMPER and delete the file by hand.

?FAILED TO FIND MAGTAPE, [reason]

Description: DUMPER cannot open the magtape for the specified reason.

Suggested User Response: Fix the problem specified by [reason] and try again.

?FAILED TO MOUNT MTA, Device already assigned to another job

Description: You used the TAPE command to specify a device that is not assigned to you.

Suggested user Response: Specify another device.

?FAILED TO OPEN MAGTAPE ?DEVICE IS NOT ON-LINE TRY AGAIN?

Description: The tape you are trying to access is not on line.

Suggested User Response: Type Y(yes) if you want to try again or N(no) if you do not. If you have further questions, contact the operator.

%Failed to restore [file] was not retrieved because: [reason]

Description: You requested that a file be retrieved from an ARCHIVE or MIGRATE tape. The operator was not able to restore the file for the reason specified.

Suggested User Response: Fix the problem specified by [reason] and try again.

%File author differs for file [file]

Description: A CHECK command found a file whose author differs from that stored on tape.

%File [file] continued from previous reel has missing page(s)

Description: The reel was changed in the middle of saving a file. In attempting to do a restore, you discovered that page numbers are missing or they do not match.

Suggested User Response: Restore the file from another tape.

THE DUMPER PROGRAM

%File [file] continued on next reel

Description: You attempted to access a file that is split between two tapes. This message informs you that the remainder of the file is on the next reel. DUMPER performs a EWIND SWITCHING to the next volume.

Suggested User Response: Ignore this message

?File [file] deleted while superseding

Description: The disk file has a higher generation number than the tape file that replaces it. The disk file is deleted.

?File [file] has page(s) missing

Description: You attempted to restore a file, and the number of pages restored did not match the number of pages indicated by the FDB.

Suggested User Response: Verify that all pages were restored.

%File last-writer differs for file [file]

Description: A CHECK command found that the last writer of the specified file differs from that on tape.

?File [file] not dumped because:
[reason]

Description: The file was not written to tape for the reason specified.

Suggested User Response: Fix the problem specified by [reason] and try the SAVE again.

%File not found - [file]

%No files dumped

Description: You attempted to save a nonexistent file.

Suggested User Response: Retype the file spec or check your directory to make sure the file has not been deleted or renamed.

%File trailer missing for file [file]

Description: The record following the data records is missing for the specified file on tape.

%Files being restored without tape information

Description: You are restoring files from an ARCHIVE tape.

Suggested User Response: Use RETRIEVE to restore the files.

?First nonspace character not a digit

Description: You attempted to SET BLOCKING-FACTOR to a specification other than a digit.

Suggested User Response: Retype the command using a decimal number.

THE DUMPER PROGRAM

%Invalid account for file [file],
system default used

Description: You attempted to restore a file in which the account number on tape is different from your account on disk.

Suggested User Response: The message is only a reminder, and no action is necessary. The file is restored. Use the ACCOUNT SYSTEM-DEFAULT command to suppress this.

?INVALID DATE AND TIME

Description: You entered the date or time in an incorrect format.

Suggested User Response: Enter the date in any standard American or European format. Enter the time in hh:mm am/pm format or in 24-hour format.

?INVALID FILE GROUP DESCRIPTOR

Description: DUMPER failed to get a JFN when it needed to default the destination file spec.

Suggested User Response: Try the operation again.

?INVALID FILESPEC, No such file type

Description: You typed an invalid filename with the INITIAL command.

Suggested User Response: Check the files on tape or directory and make sure none has been deleted. Retype the command with the correct filename.

?Invalid magtape designator, [reason]

Description: You specified an invalid device name with the TAPE command to DUMPER.

Suggested User Response: Retype the command using the appropriate device or logical name, or return to TOPS-20 command level to MOUNT a tape or ASSIGN a tape to your job.

NOTE

If you define the logical name MTA-DUMPER: as a magnetic tape device name (such as MTA2:), you do not need to type the TAPE command after entering DUMPER.

?INVALID NUMBER

Description: You specified an invalid number of tapes with the SAVE/INCREMENTAL command.

Suggested User Response: Specify a valid number.

THE DUMPER PROGRAM

?INVALID TIME FORMAT

Description: You used one of the BEFORE or SINCE Status-Setting commands without an appropriate time format (for example, 30-May-79am).

Suggested User Response: Retype the command using any acceptable format or no time at all (default = 00:00:01).

?JSYS ERROR: STRUCTURE IS NOT MOUNTED

Description: You tried to do a transfer to or from a structure that is not mounted.

Suggested User Response: Respecify a mounted structure or exit to TOPS-20 and mount the desired structure.

?Labeled tapes can be switched only to volume 1

Description: You specified a volume number other than 1 with REWIND SWITCHING, using a labeled tape.

Suggested User Response: Rewind to the first volume of the tape set.

?Labeled tapes must be MOUNTed

Description: You attempted to use the TOPS-20 command ASSIGN to mount a labeled tape.

Suggested User Response: Use the TOPS-20 command MOUNT TAPE name:.

%Log file not available, [file]

Description: DUMPER cannot open the specified log file.

Suggested User Response: Use another file.

?MAGTAPE data error record [n]

Description: DUMPER encountered a data error while reading the magtape.

Suggested User Response: Verify that the affected file is correct.

?MAGTAPE record length error record [n] [result]

Description: While reading the magtape, DUMPER read a record with an incorrect length.

Suggested User Response: Verify that the affected file is correct.

?MAGTAPE unknown error record [n] [result]

Description: A read failed for the magtape, but the monitor did not return any error bits.

Suggested User Response: Verify that the file is correct. If the error persists, contact your Software Specialist or send a Software Performance Report (SPR) to DIGITAL.

THE DUMPER PROGRAM

%No interruptable command in progress

Description: You typed CTRL/E when no command action was in progress (for example, in response to DUMPER> or a request such as \$TAPE file spec).

Suggested User Response: Press RETURN. DUMPER responds with its prompt or with the last request typed.

%No such device - [device]

Description: You typed an unassigned device name with the TAPE command and then attempted to do a transfer.

Suggested User Response: Retype TAPE, including the correct device or logical name. Then retype the transfer command.

%No such file type - [file]

Description: You typed an invalid filename with the SAVE command.

Suggested User Response: Check the directory to make sure the desired file has not been deleted or renamed. Retype the SAVE command specifying the correct filename.

?NO TAPE DEVICE SPECIFIED YET

Description: You gave the UNLOAD command without specifying a tape.

Suggested User Response: Use the TAPE command to first specify a tape.

?NOT A DECIMAL NUMBER

Description: You specified something other than a decimal number with DENSITY, SET TAPE-NUMBER, SET BLOCKING-FACTOR, SKIP, or REWIND SWITCHING.

Suggested User Response: Retype the command using an appropriate decimal number.

?NOT A DEFINED COMMAND

Description: You mistyped a command or tried to use a command that does not exist in DUMPER.

Suggested User Response: Enter a valid DUMPER command.

?NOT A SWITCH OR FILESPEC, [error]

Description: You typed incorrect data at a point where DUMPER required a switch or a file spec.

Suggested User Response: The reason specified by [error] describes the problem. Retype the correct command.

THE DUMPER PROGRAM

?NOT CONFIRMED

Description: You typed an argument with a command that does not take an argument.

Suggested User Response: Retype the command with no argument.

?Not enough monitor table space for current BLOCKING-FACTOR

Description: You specified a record that was larger than the available buffer space in the monitor.

Suggested User Response: Use a smaller BLOCKING-FACTOR value.

?Carriage return required

Only one filespec allowed for archival or migration runs

Description: (Privileged User Only) You typed more than one file spec in a SAVE command using /ARCHIVE or /MIGRATE.

Suggested User Response: Type only one file spec when you use these switches.

?OUTPUT FILESPEC INVALID, Invalid wildcard designator

Description: You included an invalid wildcard in a file spec, or specified a wildcard where it is not appropriate.

Suggested User Response: Retype the [file] with an acceptable wildcard or no wildcard.

%Partial file [file] skipped

Description: At the beginning of a tape, DUMPER found a file that continued from a previous tape.

?PDL overflow trap in DUMPER

?PC=[n]

Description: A stack overflow trap occurred at the specified PC.

Suggested User Response: This is not expected to happen. If it does, contact your Software Specialist or send a Software Performance Report (SPR) to DIGITAL.

%Please mount tape [valid]

Description: (Privileged User Only) It is necessary to mount the specified tape volume in order to RETRIEVE a requested archived or migrated file.

Suggested User Response: Locate the appropriate reel and mount it.

%Retrieve aborted

Description: (Privileged User Only) You did not find the requested file on either set of ARCHIVE or MIGRATE tapes

Suggested User Response: Inform the requestor.

THE DUMPER PROGRAM

?RETRIEVE requires WHEEL or OPERATOR capabilities enabled

Description: You attempted to use the RETRIEVE command when you do not have WHEEL or OPERATOR capabilities enabled.

Suggested User Response: Enable these capabilities or do not use RETRIEVE. If you do not have WHEEL or OPERATOR privileges, you might have intended to use RESTORE.

?Sequence error, record [n], continuing.

Description: During a restore, DUMPER encountered a record sequence number that is not one greater than the previous record. DUMPER resets the internal sequence counter to the new number, and continues with the current operation. If the sequence number is irregular, you may receive a second message when the numbers regain their previous value.

Suggested User Response: Carefully check the file being restored at the time the message was issued. If the file contains errors, you should attempt to restore the file from the backup copy of the tape.

%Structure not mounted, skipping file [file]

Description: (Privileged User Only) You attempted to retrieve a file that was archived or migrated as a structure not currently mounted.

Suggested User Response: Requeue the request and process the next one. Mount the appropriate structure so that DUMPER can locate the file when the request is processed again.

?Switch combination invalid

Description: You specified an incremental switch with /ARCHIVE or /MIGRATE. The incremental switches can be used only with SAVE. /ARCHIVE and /MIGRATE cannot be used with an incremental switch.

Suggested User Response: If you are archiving or migrating files, do not use any form of /INCREMENTAL. If you wish to do an incremental save, use SAVE followed by an incremental switch.

?Tape count must be greater than 0

Description: (Privileged User Only) You specified a 0 or negative number with /INCREMENTAL: n.

Suggested User Response: Retype the command using a positive decimal number.

%Tape does not start with header, continuing...

Description: You did not position the tape at the beginning of the saveset before doing a RESTORE.

Suggested User Response: Type CTRL/E. DUMPER responds with Interrupting... and its prompt. Type REWIND, and then type RESTORE.

THE DUMPER PROGRAM

?Tape error prevents proper positioning of tape for new saveset
\$Mark this tape as full.

Description: When you tried to append archival or migration data to an old tape, DUMPER could not find the proper tape location.

Suggested User Response: Start with a new tape and mark this one as full.

?TAPE FULL. PLEASE MARK IT.

Description: DUMPER reached the end of the tape during an ARCHIVAL or RETRIEVAL operation.

Suggested User Response: Mark the tape as full, and continue with a new tape.

%Tape has file(s) missing

Description: While DUMPER was reading a tape, the file sequence number did not increment as expected.

Suggested User Response: Verify that there are no missing files on the tape.

%Tape is format version [n], not current

Description: You attempted to read or write a tape that was written with an earlier than the current version of DUMPER.

Suggested User Response: Use the FORMAT (VERSION NUMBER IS) n command to specify which version the tape was written in.

%Tape is write protected, Type <CR> when ready to try again

Description: You attempted to write to a tape without a write ring in the back.

Suggested User Response: Remount the tape and specify that the tape is to be write-enabled.

%Tape may be wrong format

Description: There is a read data error at the beginning of the tape. The tape may have been written in INTERCHANGE format.

Suggested User Response: Rewind the tape and use the INTERCHANGE command. If this is not successful, mount another tape.

Tape specification needed,
\$Tape filespec

Description: You tried to access a magnetic tape, but did not type the TAPE command. DUMPER postpones the command you used and tries to use the logical name, MTA-DUMPER:. That name, however, was not defined with a TOPS-20 ASSIGN, TMOUNT, or DEFINE command.

Suggested User Response: Type the name of the magnetic tape drive you are using, followed by RETURN, or return to TOPS-20 command level and define the logical name MTA-DUMPER:.

THE DUMPER PROGRAM

%Tapes not in order, cannot restore continued file [file]
Mount correct tape and type CONTINUE

Description: You attempted to restore a file continued on another tape. The next tape mounted does not contain the continued file.

Suggested User Response: Locate and mount the correct tape.

%TAPES NOT IN ORDER, continuing

Description: During scanning or skipping, the end of the tape is encountered. The next tape is not the correct one in sequence.

Suggested User Response: Locate and mount the correct tape.

?This tape has been previously used for a different DUMPER purpose.

Description: You attempted to migrate, archive, or backup files onto a tape that was previously used for a function other than the one you specified.

Suggested User Response: Use a new tape, or a tape already used for the same function as the one you are currently performing.

?TOO MANY ITEMS IN FILESPEC LIST

Description: You typed too many file specs for DUMPER to store in its tables.

Suggested User Response: Use fewer file specs in your SAVE command. If necessary, split up the files into groups and make several savesets.

?Unable to clear archive/migration request bit after migration to tape complete.

Description: (Privileged User Only) After you saved a file on tape for migration or archiving, the ARCF JSYS failed to clear the archive status.

Suggested User Response: Contact your Software Specialist or send a Software Performance Report (SPR) to DIGITAL.

?UNABLE TO SET DENSITY, JOB DEFAULT USED

Description: DUMPER was not able to set the density of the tape to your specification. Therefore, it uses the default of the current job.

Suggested User Response: Accept the selected default, or pick a density that is allowed.

%UNABLE TO SET TAPE DATA MODE, JOB DEFAULT USED

Description: You attempted to set the tape data mode to the INDUSTRY-COMPATIBLE mode of 36-bits. DUMPER is not able to set your tape, and reverts to CORE-DUMP mode.

THE DUMPER PROGRAM

?Unexpected error reading tape

Description: A read failed for a magtape. There was no known error code returned by the monitor.

Suggested User Response: Try the operation again. If the error persists, contact your Software Specialist or send a Software Performance Report (SPR) to DIGITAL.

?Unexpected memory execute trap in DUMPER

?VA=[n]

?PC=[n]

Description: A fatal address fetch occurred at the given PC and virtual address.

Suggested User Response: Contact your Software Specialist or send a Software Performance Report (SPR) to DIGITAL.

?Unexpected memory read trap in DUMPER

?VA=[n]

?PC=[n]

Description: A read failure occurred at the given PC and virtual address.

Suggested User Response: Contact your Software Specialist or send a Software Performance Report (SPR) to DIGITAL.

?Unexpected memory write trap in DUMPER

?VA=[n]

?PC=[n]

Description: A write failure occurred at the given PC and virtual address.

Suggested User Response: Contact your Software Specialist or send a Software Performance Report (SPR) to DIGITAL.

%UNRECOGNIZED RECORD TYPE ON INTERCHANGE TAPE, RECORD SKIPPED

Description: DUMPER attempted to read a record of an unknown type from an interchange tape.

Suggested User Response: If this is a tape written under TOPS-10, REWIND and use INTERCHANGE. Retype the command that produced this message.

%Use the DISMOUNT TAPE command to unload MOUNTed tapes

Description: You attempted to use UNLOAD to DUMPER. UNLOAD does not work if you mounted a tape using MOUNT to TOPS-20.

Suggested User Response: Exit from DUMPER and type the TOPS-20 command DISMOUNT TAPE device name:.

THE DUMPER PROGRAM

?VOLUME NUMBER MUST BE POSITIVE

Description: You typed a negative number with REWIND SWITCHING.

Suggested User response: Retype the command using the number of the volume in the tape set.

?WHEEL or OPERATOR capabilities required

Description: You attempted to use a DUMPER command (such as SAVE/ARCHIVE) to which you do not have access.

Suggested User Response: If you have WHEEL or OPERATOR capabilities, return to TOPS-20 and type ENABLE. The system responds with a dollar sign (\$) prompt. If you do not have such capabilities, do not use the DUMPER command that caused the message.

%Write error on tape, record [n], writing duplicate record

Description: Due to areas on the tape where data cannot be read or written, DUMPER encountered an error in the record just saved on tape. Therefore, it writes a duplicate record immediately following the invalid record, and prints the message above.

Suggested User Response: If there are many errors, it is wise to replace the tape. In some cases, if there are only a few errors, the tape is probably dirty. On newer drives, repeated saving procedures will clean it and reduce errors. On older drives, use a tape cleaning device.

?Wrong tape mounted, cannot retrieve continued file

Description: You attempted to restore a file that was split between two tapes when saved. You restored the first portion of the file and mounted a second tape. The remainder of the file is not on this tape.

Suggested User Response: Locate the appropriate tape, mount it, run DUMPER and restore the remainder of the file.

?WRITE ERROR ON TRAILER RECORD!

?this [type] tape is valid for the current and all previous runs

?But, it should NOT be used for any future runs!!!

?***** MARK THIS TAPE AS FULL *****

Description: A write error occurred while DUMPER was finishing an archival or migration run.

Suggested User Response: Use another tape for future archival or migrations.

?YES or NO only

Description: You typed something other than Y for yes or N for no in response to a question issued by DUMPER.

Suggested User Response: Respond with Y or N.

CHAPTER 8

THE PLEASE PROGRAM

8.1 INTRODUCTION

The PLEASE program allows you to communicate with the operator. You can use this program to:

- ask the operator to perform a specific task
- ask the operator for information concerning your job
- supply the operator with information regarding your job.

Commands and switches within the program allow you to vary the type of message processing. The operator may reply to the message, depending on the command or switch you give.

8.2 RUNNING PLEASE

You can give commands to PLEASE at either TOPS-20 command level or at PLEASE program command level. At TOPS-20 command level (Section 8.2.1), you type PLEASE and then type your message preceded by a node name and/or any switches. At program command level (Section 8.2.2), you type PLEASE and then press the RETURN key. After the prompt, you can type your message preceded by a command and a node name and/or switch.

NOTE

The PLEASE command that you use in a batch job does not function like the timesharing PLEASE program. For information on using the PLEASE command in a batch job, refer to the TOPS-10/TOPS-20 Batch Reference Manual.

THE PLEASE PROGRAM

8.2.1 Running PLEASE at TOPS-20 Command Level

To run PLEASE at TOPS-20 command level, type PLEASE after the TOPS-20 prompt @, type your message text on the same line, and press RETURN. In this case, the program assumes that you are waiting for a reply. For example:

```
@PLEASE What's holding up my print job(CTRL/V) ?(RET)
[Message sent at 13:31:40 waiting for operator response]
```

NOTE

If you want to include special characters, such as a question mark or an ESCAPE, in your message, you must precede the character with CTRL/V.

You can send a message that exceeds one line. To do this, type a hyphen (-) after the last word in the first line of your message, and press RETURN. Continue your message on the next line and any other lines following it, ending each line to be continued with a hyphen. To send your message to the operator, press RETURN after the last line of your message.

After you press RETURN, the program outputs the time you sent the message and a reminder that you are waiting for the operator to reply. You do not receive the TOPS-20 prompt after this notification; the program assumes that you are waiting for a response from the operator. When you receive the response, you are returned to TOPS-20 command level. For example:

```
13:31:55 From operator terminal 3:
=>Had to change ribbon...your listing printing now.
@
```

The response contains the time that the operator sent the reply.

You can exit from PLEASE by typing two CTRL/C's and still receive the operator's reply. However, your reply from the operator is lost unless you do one of the following:

- Type the CONTINUE command.
- Type the RESET command.
- Run another program immediately after you send your PLEASE message.

If there is no operator in attendance, PLEASE informs you and immediately returns you to TOPS-20 command level without waiting for a response. Your message is queued until an operator is available to reply. For example:

```
@PLEASE Has my new tape been mounted on MTA1?(RET)
[Message sent at 10:10:33 there is no operator in attendance]
@
```


THE PLEASE PROGRAM

When you run PLEASE at TOPS-20 command level, you can include an optional node name and/or one or more switches in the command line.

Node Names

A node name assures that only operators at a remote node receive your message. If an operator is in attendance at the node you specify, the message is sent there. Otherwise, the message is sent to the central site operator. To send a PLEASE message to a node, type PLEASE, the node name followed by two colons, the message, and press RETURN. You must leave a space between the word PLEASE and the node name. A space is optional between the second colon and the beginning of your message. You send a message to a node as in the following example:

```
@ PLEASE DN200:: Cancel my Batch Job. (RET)
[Message sent at 11:23:34 waiting for operator response]
```

Switches

You can type a switch after the node name in the command line. These switches determine the type of message processing. To include switches in the command line, include them immediately before your message.

/MESSAGE

Type /MESSAGE to indicate to the operator that you do not expect a reply to your message. For example:

```
@ PLEASE /MESSAGE This is only a test. (RET)
[Operator notified at 10:46:35]
@
```

After you press RETURN, PLEASE gives you the time that the operator received your message. If you do not specify /MESSAGE in the command line, the program assumes that you expect a reply from the operator.

/NOWAIT

Type /NOWAIT to send a message to the operator and immediately return to TOPS-20 command level. You do not wait for a reply to your message. When it is convenient for the operator to reply, you receive the response. If you do not specify /NOWAIT in the command line, the program assumes that you are waiting for a reply from the operator and does not return you to TOPS-20 command level.

For example, if you wish to send a message to the operator at node 4097 and do not wish to wait for the reply, the command line looks like the following:

```
@ PLEASE 4097:: /NOWAIT Any stand-alone time open tomorrow (CTRL/\) ? (RET)
[Message sent at 10:47:27]
@
```

THE PLEASE PROGRAM

8.2.2 Running PLEASE at Program Command Level

To run PLEASE at program command level, type PLEASE after the TOPS-20 prompt @, and press RETURN. The program responds with the prompt as follows:

```
@PLEASE RET  
PLEASE>
```

This prompt is for you to type a program command, with or without an optional node name, a switch, and message text.

Commands

There are four commands you can use when you run PLEASE at program command level. You use two of them, SEND and MESSAGE, in a PLEASE command line to send a message to the operator. You use the other two, CANCEL and EXIT, for non-message functions.

The SEND Command

Type SEND after the PLEASE prompt to send a message to the operator and indicate that you expect a reply to your message. The command line can include an optional node name. (Refer to Section 8.2.1.) You can also include /NOWAIT to indicate to the operator that you are not waiting for the reply to your message; you return to program command level after you press RETURN. After the command, the optional node name and the optional switch, type your message text. Remember to use a hyphen if you wish to send a message that exceeds one line.

The syntax for the SEND command line is as follows:

```
@PLEASE RET  
PLEASE>SEND [node::] [/NOWAIT] text... RET
```

For example, you wish to send a message to node DN20 and wait for the reply:

```
@PLEASE RET  
PLEASE>SEND DN20:: Waiting for answer to tape question. RET  
[Message sent at 10:56:49 waiting for operator response]
```

In this case, you do not receive the PLEASE prompt until the operator replies or unless you use /NOWAIT in the command line.

You have another option with the SEND command. Instead of typing your message text on the same line (with an optional node name and/or switch), you can press RETURN before you type the message; for example:

```
@PLEASE RET  
PLEASE>SEND DN20:: RET  
Enter Text and Terminate with ^Z  
Waiting for an answer to tape question. RET  
Do you need more information? RET  
CTRL/Z  
[Message sent at 13:44:39 waiting for operator response]
```

Note that in this case, you type CTRL/Z after you complete the message.

THE PLEASE PROGRAM

The MESSAGE Command

Type MESSAGE after the PLEASE prompt to send a message to the operator and indicate that you do not expect a reply to your message. The command line can include an optional node name. When you type the text of your message, remember to use a hyphen if the message exceeds one line.

The syntax for the MESSAGE command line is as follows:

```
@PLEASE RET  
PLEASE>MESSAGE [node:] text... RET
```

For example, you wish to send a test message to the operator at your installation:

```
@PLEASE RET  
PLEASE MESSAGE This is just a test. RET  
[Operator notified at 11:06:35]  
PLEASE>
```

You have another option with the MESSAGE command. Instead of typing your message text on the same line (with an optional node name), you can press RETURN before you type the message; for example:

```
@PLEASE RET  
PLEASE>MESSAGE 2116: RET  
Enter Text and Terminate with ^Z  
This message goes to an operator at node 2116. RET  
It is just a test. RET  
[CTRL/Z]  
[Operator notified at 14:01:15]  
PLEASE>
```

In the case of both the SEND and MESSAGE command lines, the program returns with the PLEASE prompt after it processes your message. You can type another command line, or one of the following commands:

The CANCEL Command

Type CANCEL after the PLEASE prompt to cancel all outstanding messages your job has queued for the operator to answer.

For example, if you send three messages to the operator and then wish to cancel all of them:

```
@PLEASE RET  
PLEASE>CANCEL (all outstanding Messages) RET  
[3 Message(s) Canceled]  
PLEASE>
```

The EXIT Command

Type EXIT to exit from the PLEASE program and return to TOPS-20 command level. For example:

```
@PLEASE RET  
PLEASE>EXIT RET  
@
```

THE PLEASE PROGRAM

8.3 PLEASE MESSAGES

The following is an alphabetical list of the most common PLEASE messages. A warning message is preceded by a percent sign (%). Fatal errors are preceded by a question mark (?). A description is supplied for each message, along with a suggested user response where applicable.

? GLXINI - Unable to obtain run-time system

Description: You attempted to run PLEASE from a directory that does not include GLXLIB.EXE.

Suggested User Response: Either run PLEASE from a directory that includes GLXLIB.EXE or put a copy of it in the directory that contains PLEASE.EXE.

? Invalid PLEASE command specified

Description: You typed an invalid PLEASE command after the program prompt.

Suggested User Response: Type one of the four valid PLEASE commands after the program prompt.

? Invalid PLEASE switch specified

Description: You typed an invalid PLEASE switch in the command line.

Suggested User Response: Type one of the two valid PLEASE switches in the command line.

[Message sent at [time]] (waiting for operator response)
or
(there is no operator in attendance)

Description: PLEASE sent your message to the operator at the specified time. If you asked to wait for a response and the operator is in attendance, PLEASE types the first option. If there is no operator in attendance, PLEASE types the second option.

% Message truncated - text exceeded buffer capacity

Description: You typed a message longer than PLEASE can handle.

Suggested User Response: Type only messages that are 2,000 characters or less.

[Operator notified at [time]]

Description: PLEASE sent your message to the operator at the specified time. You did not request a response, so PLEASE does not wait for one.

THE PLEASE PROGRAM

? There are no outstanding messages for this program

Description: You attempted to cancel all of your queued messages to the operator and there were none outstanding.

? Unrecognized response from ORION

Description: PLEASE received a message from ORION, the operator interface, that it could not understand. Your installation may be running an old version of PLEASE.

Suggested User Response: Contact your Software Specialist or send a Software Performance Report (SPR) to DIGITAL.

INDEX

/A switch,
 CREF, 5-4
 FILCOM, 4-3, 4-11
 ABEFORE command,
 DUMPER, 7-8, 7-51
 Absolute break, 6-4
 ACCOUNT command,
 DUMPER, 7-8, 7-14, 7-51
 Action commands,
 DUMPER, 7-7, 7-23, 7-24
 Activating an edit, 6-23
 Active edit, 6-20
 ALGOL compiler source files,
 5-1
 ALGOL CREF listing example,
 5-9
 /ALL switch,
 RDMAIL, 3-4, 3-5
 Allocation,
 status of tape drive, 7-3
 Tape drive, 7-3
 .ALTER pseudo-op, 6-21
 /APPEND switch,
 MAKLIB, 6-7, 6-27
 /APPEND switch example,
 MAKLIB, 6-9
 /APPEND switch figure,
 MAKLIB, 6-8
 Appending modules to
 libraries, 6-7
 ARCHIVE command, 7-31
 /ARCHIVE switch,
 DUMPER, 7-43, 7-48, 7-55
 Archived and migrated files,
 Restoring, 7-50
 Retrieving, 7-50
 Archived files, 7-1
 Archiving,
 Marking files for, 7-31
 Archiving files, 7-3, 7-43,
 7-47, 7-48
 Archiving marked files,
 7-46
 Arguments,
 MAKLIB switch, 6-3
 ASCII file comparison
 switches, 4-3
 ASCII mode file comparison,
 4-3
 ASCII output file format,
 4-4
 ASCII output file header,
 4-4
 ASCII output file header
 example, 4-4
 ASCII text blocks, 6-6

ASINCE command,
 DUMPER, 7-8, 7-9, 7-51
 Assembler,
 MAKLIB MACRO, 6-24, 6-25
 Assembler source files,
 MACRO, 5-1
 ASSEMBLER source files,
 MACRO, 4-11
 Assembler source files,
 MACRO, 4-3
 Assembly language source
 files,
 Comparing, 4-3, 4-11
 Assigning a tape drive, 7-4
 Assigning a tape drive
 example, 7-4
 .ASSOCIATED pseudo-op, 6-21

 /B switch,
 CREF, 5-4
 FILCOM, 4-3, 4-11
 Backing up files, 7-44
 Backup,
 Restoring files from,
 7-46
 Backup files, 7-1
 Batch job,
 MAIL used in a, 2-6
 Batch job control file
 example, 2-6
 Batch job PLEASE command,
 8-1
 BEFORE command,
 DUMPER, 7-8, 7-51
 Beginning of tape,
 Positioning at, 7-21
 Binary file comparison
 switches, 4-8
 Binary file types, 4-2
 Binary files,
 Comparing, 4-7
 Listing, 4-7
 Binary output file format,
 4-8
 Binary output file header,
 4-8
 Blank lines in file
 comparison, 4-3
 Blocking, 7-15, 7-16
 Blocking factors, 7-15
 BLOCKING-FACTOR command,
 DUMPER SET, 7-16
 Blocks,
 ASCII text, 6-6

INDEX (CONT.)

Blocks (Cont.)
 DUMPER File Descriptor,
 7-28
 END, 6-4
 Inserting INDEX, 6-19
 REQUEST, 6-6
 REQUIRE, 6-6
 SYMBOL, 6-20
 TRACE, 6-5, 6-20, 6-21,
 6-26
 Blocks format,
 TRACE, 6-41
 BUILD command, 3-1

 /C switch,
 CREF, 5-4
 FILCOM, 4-3, 4-11
 CANCEL command,
 PLEASE, 8-5
 CANCEL MOUNT-REQUEST
 command, 7-50
 Canceling outstanding
 PLEASE messages, 8-5
 Capabilities in DUMPER,
 Enabling, 7-1, 7-42
 Capabilities in MAIL,
 Enabling, 2-5
 Capabilities in RDMAIL,
 Enabling, 3-2
 Change bars,
 FILCOM, 4-4, 4-7, 4-11
 Characters,
 .CRF file control, 5-15
 .CRF input file control,
 5-14, 5-16, 5-17
 Radix-50, 6-21
 SIXBIT, 6-21
 CHECK command,
 DUMPER, 7-27, 7-51
 Checking,
 Continuing, 7-19
 CHECKSUM command,
 DUMPER, 7-8, 7-18, 7-51
 Checksum during SAVE,
 Specifying a, 7-18
 COBOL compiler source files,
 5-1
 Code and symbol table in
 modules, 6-25
 Code ending point,
 MAKLIB, 6-23
 Code format,
 .FIX files, 6-43
 Code in modules,
 Deleting, 6-22
 Inserting, 6-22

 Code insertion restriction,
 6-22
 Code segment example, 6-23
 Command,
 ARCHIVE, 7-31
 Batch job PLEASE, 8-1
 BUILD, 3-1
 CANCEL MOUNT-REQUEST,
 7-50
 COMPILE, 5-1, 5-2
 CONTINUE, 8-2
 Continuing an interrupted,
 7-30
 DEFINE, 1-2, 1-3
 DUMPER ABEFORE, 7-8, 7-51
 DUMPER ACCOUNT, 7-8, 7-14,
 7-51
 DUMPER ASINCE, 7-8, 7-9,
 7-51
 DUMPER BEFORE, 7-8, 7-51
 DUMPER CHECK, 7-27, 7-51
 DUMPER CHECKSUM, 7-8,
 7-18, 7-51
 DUMPER CONTINUE, 7-30, 7-52
 DUMPER CREATE, 7-43, 7-46,
 7-52
 DUMPER CTRL/E, 7-29, 7-52
 DUMPER DENSITY, 7-8, 7-15,
 7-52
 DUMPER DIRECTORIES, 7-8,
 7-11, 7-52
 DUMPER DISMOUNT TAPE, 7-5
 DUMPER EOT, 7-20, 7-21,
 7-52
 DUMPER EXIT, 7-7, 7-52
 DUMPER FILES, 7-8, 7-11,
 7-52
 DUMPER FORMAT, 7-8, 7-16,
 7-53
 DUMPER HELP, 7-7, 7-53
 DUMPER INDUSTRY, 7-8,
 7-16, 7-53
 DUMPER INITIAL, 7-8, 7-19,
 7-53
 DUMPER INTERCHANGE, 7-8,
 7-16, 7-53
 DUMPER LIST, 7-8, 7-12,
 7-53
 DUMPER MBEFORE, 7-8, 7-9,
 7-53
 DUMPER MOUNT TAPE, 7-4,
 7-6
 DUMPER MSINCE, 7-8, 7-9,
 7-53
 DUMPER NO CHECKSUM, 7-51
 DUMPER NO CREATE, 7-52
 DUMPER NO DIRECTORIES,
 7-11, 7-52

INDEX (CONT.)

Command (Cont.)

DUMPER NO FILES, 7-12,
7-52
DUMPER NO INDUSTRY, 7-53
DUMPER NO INTERCHANGE,
7-53
DUMPER NO LIST, 7-13,
7-53
DUMPER NO SILENCE, 7-13,
7-55
DUMPER PARITY, 7-8, 7-15,
7-53
DUMPER PRINT, 7-29, 7-54
DUMPER PROTECTION, 7-8,
7-14, 7-54
DUMPER QUIT, 7-7, 7-54
DUMPER RESTORE, 7-23,
7-26, 7-43, 7-46, 7-54
DUMPER RETRIEVE, 7-43,
7-50, 7-54
DUMPER REWIND, 7-20, 7-21,
7-54
DUMPER REWIND SWITCHING,
7-21
DUMPER SAVE, 7-23, 7-24,
7-55
DUMPER SET
BLOCKING-FACTOR, 7-8,
7-15, 7-16, 7-55
DUMPER SET TAPE-NUMBER,
7-8, 7-19, 7-55
DUMPER SILENCE, 7-8, 7-13,
7-55
DUMPER SINCE, 7-8, 7-9,
7-55
DUMPER SKIP, 7-22, 7-56
DUMPER SKIP n, 7-20
DUMPER SSNAME, 7-8, 7-18,
7-56
DUMPER SUPERSEDE, 7-8,
7-10, 7-56
DUMPER TAPE, 7-8, 7-14,
7-56
DUMPER UNLOAD, 7-20, 7-23,
7-56
GET, 2-3
INFORMATION
ARCHIVE-STATUS, 7-31
INFORMATION MAIL, 2-4,
3-2
INFORMATION SYSTEM-STATUS,
7-3
INFORMATION VOLUMES, 7-6
LOAD-class, 5-1
PLEASE CANCEL, 8-5
PLEASE EXIT, 8-5
PLEASE MESSAGE, 8-5
PLEASE SEND, 8-4
REENTER, 2-3

Command (Cont.)

REFUSE SYSTEM-MESSAGES,
3-2
RESET, 8-2
Running CREF as a, 5-5
SET MAIL-WATCH, 3-2
TALK, 2-4
TRANSLATE, 1-2
TYPE, 4-5, 5-6, 6-5
Command categories,
DUMPER, 7-7
Command in DUMPER,
Halting a, 7-29
Command line format,
PLEASE MESSAGE, 8-5
PLEASE SEND, 8-4
Command string format,
CREF, 5-3
FILCOM, 4-1
MAKLIB, 6-3
Command string requirements,
MAKLIB, 6-4, 6-7, 6-19,
6-25
Commands,
DUMPER Action, 7-7, 7-23,
7-24
DUMPER Status-Setting,
7-7, 7-8
DUMPER Tape-Positioning,
7-7, 7-20
DUMPER Transfer, 7-23
MAKLIB multi-line, 6-4
PLEASE, 8-1, 8-4
Comments and spaces in file
comparison, 4-3
Communicating with the
operator, 8-1
COMND.CMD file, 3-2
Comparing ASCII files, 4-3
Comparing ASCII files
example, 4-5, 4-6
Comparing assembly language
source files, 4-3, 4-11
Comparing binary files, 4-7
Comparing binary files
example, 4-9
Comparing executable files
example, 4-10
COMPILE command, 5-1, 5-2
COMPILE command example,
5-1, 5-2
Compiler source files,
ALGOL, 5-1
COBOL, 5-1
FORTRAN, 5-1
Compiling a program, 5-1,
5-2
Compiling ALGOL source
files example, 5-9

INDEX (CONT.)

- Compiling FORTRAN source
 - files example, 5-8
- Compiling MACRO source
 - files example, 5-6
- CONTINUE command, 8-2
 - DUMPER, 7-30, 7-52
- Continuing an interrupted
 - command, 7-30
- Continuing checking, 7-19
- Continuing restoring, 7-19
- Continuing saving, 7-19
- Control characters,
 - .CRF file, 5-15
 - .CRF input file, 5-14, 5-16, 5-17
- Control pseudo-op, 6-20
- CREATE command,
 - DUMPER, 7-43, 7-46, 7-52
- Creating a saveset, 7-24
- Creating .CRF files, 5-1
- Creating .CRF files example, 5-2
- Creating .FIX files, 6-20
- Creating libraries, 6-9
- CREF,
 - Running, 5-1
 - Running other programs from, 5-6
- CREF /A switch, 5-4
- CREF as a command,
 - Running, 5-5
- CREF as a program,
 - Running, 5-5
- CREF /B switch, 5-4
- CREF /C switch, 5-4
- CREF command string format, 5-3
- CREF /D switch, 5-4
- CREF destination file spec, 5-3
- CREF error messages, 5-10
- CREF error status codes, 5-14
- CREF example, 5-6
- CREF file access errors, 5-13
- CREF /H switch, 5-4
- CREF indirect file, 5-5
- CREF informational messages, 5-10
- CREF /K switch, 5-4
- CREF /M switch, 5-4
- CREF MACRO symbols and references example, 5-18
- CREF message variables, 5-10
- CREF /O switch, 5-4
- CREF /O switch example, 5-5
- CREF octal codes, 5-13
- CREF /P switch, 5-4
- CREF program, 5-1
- CREF /R switch, 5-4
- CREF /S switch, 5-4
- CREF source file spec, 5-3
- /CREF switch, 5-1, 5-2
- /CREF switch example, 5-1, 5-2
- CREF SWITCH.INI file, 5-5
- CREF switches, 5-3
- CREF /T switch, 5-4
- CREF technical notes, 5-14
- CREF /W switch, 5-4
- CREF warning messages, 5-10
- CREF /Z switch, 5-4
- .CRF file control
 - characters, 5-15
- .CRF files,
 - Creating, 5-1
- .CRF input file control
 - characters, 5-14, 5-16, 5-17
- .CRF input file format, 5-14
- Cross-reference listings, 5-1
 - Producing, 5-2
 - tables in, 5-2
- CTRL/E command,
 - DUMPER, 7-29, 7-52
- CTRL/H,
 - MAIL, 2-1, 2-4
- CTRL/V,
 - MAIL, 2-2
- CTRL/Z,
 - MAIL, 2-2
- /D switch,
 - CREF, 5-4
- Date,
 - Edit creation, 6-21
- Date and time,
 - RDMAIL, 3-3
 - Transferring files by, 7-8, 7-9
- .DATE pseudo-op, 6-6, 6-21
- Deactivating an edit, 6-23
- Debugging, 5-1, 6-20
- DEFINE command, 1-2, 1-3
- DEFINE command example, 1-3, 5-6
- Defining logical names
 - example, 1-3, 5-6, 7-4, 7-5
- /DELETE switch,
 - MAKLIB, 6-7, 6-9, 6-20, 6-27

INDEX (CONT.)

/DELETE switch example,
 MAKLIB, 6-11
 /DELETE switch figure,
 MAKLIB, 6-10
 Deleting a saveset, 7-25
 Deleting code in modules,
 6-22
 Deleting local symbols,
 6-20
 Density,
 Setting tape, 7-15
 DENSITY command,
 DUMPER, 7-8, 7-15, 7-52
 Destination file spec,
 CREF, 5-3
 DUMPER, 7-23
 FILCOM, 4-1
 MAKLIB, 6-3
 DIRECTORIES command,
 DUMPER, 7-8, 7-11,
 7-52
 Directory,
 PS:<REMARKS>, 2-5
 PS:<SYSTEM>, 2-5, 2-6
 Restoring a, 7-46
 Sending mail to a
 non-files-only, 2-5
 DIRECTORY command example,
 7-10
 Disk,
 Transferring files from
 tape to, 7-26
 Disk and tape,
 Transferring between,
 7-2
 Disk file,
 Overwriting a, 7-10, 7-27
 Disk to tape,
 Transferring files from,
 7-24
 DISMOUNT TAPE command,
 DUMPER, 7-5
 Dismounting a tape, 7-5
 DUMPER,
 Enabling capabilities in,
 7-1, 7-42
 Halting a command in,
 7-29
 Logical names in, 7-5
 Nonprivileged user in,
 7-7
 Privileged user in, 7-42
 Reading a tape from an
 earlier version of,
 7-16
 Running, 7-6
 Stopping, 7-7
 DUMPER ABEFORE command, 7-8,
 7-51
 DUMPER ACCOUNT command, 7-8,
 7-14, 7-51
 DUMPER Action commands, 7-7,
 7-23, 7-24
 DUMPER /ARCHIVE switch,
 7-43, 7-48, 7-55
 DUMPER /ARCHIVE switch
 example, 7-48
 DUMPER ASINCE command, 7-8,
 7-9, 7-51
 DUMPER BEFORE command, 7-8,
 7-51
 DUMPER BEFORE command
 example, 7-9
 DUMPER BSINCE command
 example, 7-10
 DUMPER CHECK command, 7-27,
 7-51
 DUMPER CHECK command
 example, 7-35
 DUMPER CHECKSUM command,
 7-8, 7-18, 7-51
 DUMPER CHECKSUM command
 example, 7-18, 7-34
 DUMPER command categories,
 7-7
 DUMPER CONTINUE command,
 7-30, 7-52
 DUMPER CONTINUE command
 example, 7-30
 DUMPER CREATE command, 7-43,
 7-46, 7-52
 DUMPER CTRL/E command, 7-29,
 7-52
 DUMPER DENSITY command, 7-8,
 7-15, 7-52
 DUMPER destination file
 spec, 7-23
 DUMPER DIRECTORIES command,
 7-8, 7-11, 7-52
 DUMPER DISMOUNT TAPE
 command, 7-5
 DUMPER EOT command, 7-20,
 7-21, 7-52
 DUMPER EOT command example,
 7-21
 DUMPER error messages, 7-57
 DUMPER EXIT command, 7-7,
 7-52
 DUMPER EXIT command example,
 7-37
 DUMPER FDB, 7-28
 DUMPER FDB example, 7-28
 DUMPER features, 7-2
 DUMPER File Descriptor
 Blocks, 7-28
 DUMPER FILES command, 7-8,
 7-11, 7-52

INDEX (CONT.)

DUMPER FILES command
 example, 7-12
 DUMPER FORMAT command, 7-8,
 7-16, 7-53
 DUMPER /FULL-INCREMENTAL
 switch, 7-42, 7-44,
 7-55
 DUMPER /FULL-INCREMENTAL
 switch example, 7-44
 DUMPER HELP command, 7-7,
 7-53
 DUMPER /INCREMENTAL:n
 switch, 7-42, 7-44,
 7-55
 DUMPER INDUSTRY command,
 7-8, 7-16, 7-53
 DUMPER INITIAL command, 7-8,
 7-19, 7-53
 DUMPER INTERCHANGE command,
 7-8, 7-16, 7-53
 DUMPER INTERCHANGE command
 example, 7-17
 DUMPER LIST command, 7-8,
 7-12, 7-53
 DUMPER LIST command example,
 7-13
 DUMPER MBEFORE command, 7-8,
 7-9, 7-53
 DUMPER message variables,
 7-57
 DUMPER /MIGRATE switch,
 7-43, 7-49, 7-55
 DUMPER MOUNT TAPE command,
 7-4, 7-6
 DUMPER MOUNT TAPE command
 example, 7-4, 7-5
 DUMPER MSINCE command, 7-8,
 7-9, 7-53
 DUMPER MSINCE command
 example, 7-10
 DUMPER NO CHECKSUM command,
 7-51
 DUMPER NO CREATE command,
 7-52
 DUMPER NO DIRECTORIES
 command, 7-11, 7-52
 DUMPER NO DIRECTORIES
 command example, 7-11
 DUMPER NO FILES command,
 7-12, 7-52
 DUMPER NO INDUSTRY command,
 7-53
 DUMPER NO INTERCHANGE
 command, 7-53
 DUMPER NO LIST command,
 7-13, 7-53
 DUMPER NO SILENCE command,
 7-13, 7-55
 DUMPER /NOINCREMENTAL
 switch, 7-43, 7-44,
 7-55
 DUMPER /NOUNLOAD switch,
 7-22
 DUMPER PARITY command, 7-8,
 7-15, 7-53
 DUMPER PRINT command, 7-29,
 7-54
 DUMPER PRINT command
 example, 7-29, 7-34,
 7-41
 DUMPER program, 7-1
 DUMPER PROTECTION command,
 7-8, 7-14, 7-54
 DUMPER QUIT command, 7-7,
 7-54
 DUMPER RESTORE command,
 7-23, 7-26, 7-43, 7-46,
 7-54
 DUMPER RESTORE command
 example, 7-27, 7-38
 DUMPER RETRIEVE command,
 7-43, 7-50, 7-54
 DUMPER REWIND command, 7-20,
 7-21, 7-54
 DUMPER REWIND SWITCHING
 command, 7-21
 DUMPER REWIND SWITCHING
 command example, 7-21
 DUMPER SAVE command, 7-23,
 7-24, 7-55
 DUMPER SAVE command example,
 7-24, 7-31, 7-34, 7-36,
 7-37
 DUMPER SET BLOCKING-FACTOR
 command, 7-8, 7-15,
 7-16, 7-55
 DUMPER SET TAPE-NUMBER
 command, 7-8, 7-19,
 7-55
 DUMPER SET TAPE-NUMBER
 command example, 7-20
 DUMPER SILENCE command, 7-8,
 7-13, 7-55
 DUMPER SINCE command, 7-8,
 7-9, 7-55
 DUMPER SKIP command, 7-22,
 7-56
 DUMPER SKIP command example,
 7-22, 7-41
 DUMPER SKIP n command, 7-20
 DUMPER source file spec,
 7-23
 DUMPER SSNAME command, 7-8,
 7-18, 7-56
 DUMPER Status-Setting
 commands, 7-7, 7-8

INDEX (CONT.)

DUMPER SUPERSEDE command,
 7-8, 7-10, 7-56
 DUMPER SUPERSEDE command
 example, 7-10
 DUMPER TAPE command, 7-8,
 7-14, 7-56
 DUMPER /TAPE-INFORMATION
 switch, 7-51
 DUMPER Tape-Positioning
 commands, 7-7, 7-20
 DUMPER Transfer commands,
 7-23
 DUMPER UNLOAD command, 7-20,
 7-23, 7-56
 DUMPER /VOLIDS switch, 7-5
 DUMPER warning messages,
 7-57
 DUMPER wildcards, 7-23
 DUMPER /WRITE-ENABLE switch
 example, 7-4
 DUMPER /WRITE-ENABLED
 switch, 7-4
 Dumping a file example, 4-9
 Dumping files, 4-8
 Duplicate user names, 2-1

 /E switch,
 FILCOM, 4-8, 4-11
 Edit,
 Activating an, 6-23
 Active, 6-20
 Deactivating an, 6-23
 Inactive, 6-20
 Related, 6-21
 Edit creation date, 6-21
 Edit creator initials, 6-21
 Edit ending point,
 MAKLIB, 6-23
 Edit history information,
 6-20
 Edit locations,
 changing, 6-21
 Edit name, 6-21
 .EDIT pseudo-op, 6-20, 6-21
 Edit version number, 6-21
 Editing libraries, 6-2,
 6-20
 Edits to libraries,
 Listing, 6-5
 Enabling capabilities in
 DUMPER, 7-1, 7-42
 Enabling capabilities in
 MAIL, 2-5
 Enabling capabilities in
 RDMAIL, 3-2
 END blocks, 6-4

 End-of-Tape,
 Positioning at, 7-21
 .ENDE pseudo-op, 6-20, 6-23
 .ENDI pseudo-op, 6-22, 6-23
 .ENDI pseudo-op example,
 6-23
 Entry points in libraries,
 Listing, 6-5
 EOT command,
 DUMPER, 7-20, 7-21, 7-52
 Error messages,
 CREF, 5-10
 DUMPER, 7-57
 FILCOM, 4-12
 MAIL, 2-3, 2-6
 MAKLIB, 6-28
 PLEASE, 8-6
 RDMAIL, 3-7
 Error recovery procedure,
 MAIL, 2-3
 Error status codes,
 CREF, 5-14
 Errors,
 CREF file access, 5-13
 FILCOM file access, 4-13,
 4-14
 Errors in sending mail, 2-3
 Example,
 ALGOL CREF listing, 5-9
 ASCII output file header,
 4-4
 Assigning a tape drive,
 7-4
 Batch job control file,
 2-6
 Code segment, 6-23
 Comparing ASCII files,
 4-5, 4-6
 Comparing binary files,
 4-9
 Comparing executable
 files, 4-10
 COMPILE command, 5-1, 5-2
 Compiling ALGOL source
 files, 5-9
 Compiling FORTRAN source
 files, 5-8
 Compiling MACRO source
 files, 5-6
 Creating .CRF files, 5-2
 CREF, 5-6
 CREF MACRO symbols and
 references, 5-18
 CREF /O switch, 5-5
 /CREF switch, 5-1, 5-2
 DEFINE command, 1-3, 5-6
 Defining logical names,
 1-3, 5-6, 7-4, 7-5

INDEX (CONT.)

Example (Cont.)

DIRECTORY command, 7-10
 DUMPER /ARCHIVE switch,
 7-48
 DUMPER BEFORE command,
 7-9
 DUMPER BSINCE command,
 7-10
 DUMPER CHECK command,
 7-35
 DUMPER CHECKSUM command,
 7-18, 7-34
 DUMPER CONTINUE command,
 7-30
 DUMPER EOT command, 7-21
 DUMPER EXIT command, 7-37
 DUMPER FDB, 7-28
 DUMPER FILES command,
 7-12
 DUMPER /FULL-INCREMENTAL
 switch, 7-44
 DUMPER INTERCHANGE
 command, 7-17
 DUMPER LIST command, 7-13
 DUMPER MOUNT TAPE command,
 7-4, 7-5
 DUMPER MSINCE command,
 7-10
 DUMPER NO DIRECTORIES
 command, 7-11
 DUMPER PRINT command,
 7-29, 7-34, 7-41
 DUMPER RESTORE command,
 7-27, 7-38
 DUMPER REWIND SWITCHING
 command, 7-21
 DUMPER SAVE command, 7-24,
 7-31, 7-34, 7-36, 7-37
 DUMPER SET TAPE-NUMBER
 command, 7-20
 DUMPER SKIP command, 7-22,
 7-41
 DUMPER SUPERSEDE command,
 7-10
 DUMPER /WRITE-ENABLE
 switch, 7-4
 Dumping a file, 4-9
 .ENDI pseudo-op, 6-23
 FILCOM /C switch, 4-6
 FILCOM /nL switch, 4-10
 FILCOM /nU switch, 4-10
 FILCOM /Q switch, 4-9,
 4-10
 FILCOM /U switch, 4-7
 FILCOM /W switch, 4-10
 .FIX files, 6-26
 FORTRAN CREF listing,
 5-8
 GET command, 2-3

Example (Cont.)

INFORMATION
 ARCHIVE-STATUS command,
 7-47
 INFORMATION SYSTEM-STATUS
 command, 7-3
 INFORMATION VOLUMES
 command, 7-6
 .INSERT pseudo-op, 6-23
 MACRO CREF listing, 5-6,
 5-21
 MAIL indirect file, 2-4
 MAKLIB /APPEND switch,
 6-9
 MAKLIB /DELETE switch,
 6-11
 MAKLIB edit, 6-23, 6-26
 MAKLIB /EXTRACT switch,
 6-13
 MAKLIB /FIX switch, 6-26
 MAKLIB /INDEX switch, 6-20
 MAKLIB /INSERT switch,
 6-15, 6-17
 MAKLIB /LIST switch, 6-4,
 6-9, 6-11, 6-13, 6-15,
 6-17, 6-19
 MAKLIB /LOAD switch, 6-6
 MAKLIB /MASTER switch,
 6-15, 6-17, 6-18
 MAKLIB /NOLOCALS switch,
 6-20
 MAKLIB /POINTS switch, 6-5
 MAKLIB /REPLACE switch,
 6-18
 MAKLIB /TRACE switch, 6-6
 MAKLIB /WHO switch, 6-26
 Opcode table, 5-6
 PLEASE CANCEL command,
 8-5
 PLEASE EXIT command, 8-5
 PLEASE MESSAGE command,
 8-5
 PLEASE /MESSAGE switch,
 8-3
 PLEASE /NOWAIT switch, 8-3
 PLEASE program level, 8-4
 PLEASE SEND command, 8-4
 PLEASE TOPS-20 command
 level, 8-2
 Procedure nesting table,
 5-9
 RDMAIL /ALL switch, 3-5
 RDMAIL /HELP switch, 3-4
 RDMAIL /LIST switch, 3-5
 RDMAIL
 /MESSAGE-OF-THE-DAY
 switch, 3-5
 RDMAIL /PERUSE switch,
 3-6

INDEX (CONT.)

Example (Cont.)

RDMAIL /STOP switch,
3-6
Reading another user's
mail, 3-2
REENTER command, 2-3
Running CREF as a command,
5-5
Running CREF as a program,
5-5
Sending mail to
PS:<REMARKS>, 2-5
Sending mail to
PS:<SYSTEM>, 2-5
SWITCH.INI file, 5-6
TRANSLATE command, 1-2
TYPE command, 4-5, 4-6,
5-6, 6-5
VDIRECTORY command, 7-10
.EXE file, 6-1
Executable program, 6-1
EXIT command,
DUMPER, 7-7, 7-52
PLEASE, 8-5
/EXIT switch,
MAKLIB, 6-27
/EXTRACT switch,
MAKLIB, 6-7, 6-11, 6-20,
6-27
/EXTRACT switch example,
MAKLIB, 6-13
/EXTRACT switch figure,
MAKLIB, 6-11
Extracting modules from
libraries, 6-11

FDB,
DUMPER, 7-28
Features,
DUMPER, 7-2
Fields of file specs, 1-2
Figure,
Generating an .EXE file,
6-1
Generating libraries, 6-2
MAIL/MAILER, 2-9
MAKLIB /APPEND switch,
6-8
MAKLIB /DELETE switch,
6-10
MAKLIB /EXTRACT switch,
6-11
MAKLIB /INSERT switch,
6-14, 6-15
MAKLIB /REPLACE switch,
6-17
Pseudo-op order, 6-24

FILCOM,
Generation numbers in,
4-2
Logical names in, 4-2
Running, 4-1
Stopping, 4-3
FILCOM /A switch, 4-3, 4-11
FILCOM /B switch, 4-3, 4-11
FILCOM /C switch, 4-3, 4-11
FILCOM /C switch example,
4-6
FILCOM change bars, 4-4,
4-7, 4-11
FILCOM command string
format, 4-1
FILCOM destination file
spec, 4-1
FILCOM /E switch, 4-8, 4-11
FILCOM error messages, 4-12
FILCOM file access errors,
4-13, 4-14
FILCOM file types, 4-2
FILCOM /H switch, 4-3, 4-8,
4-11
FILCOM help text, 4-3, 4-8,
4-11
FILCOM message variables,
4-12
FILCOM /nL switch, 4-4, 4-8,
4-11
FILCOM /nL switch example,
4-10
FILCOM nonstandard file
types, 4-8, 4-11
FILCOM /nU switch, 4-8,
4-11
FILCOM /nU switch example,
4-10
FILCOM octal codes, 4-13,
4-14
FILCOM program, 4-1
FILCOM /Q switch, 4-4, 4-8,
4-11
FILCOM /Q switch example,
4-9, 4-10
FILCOM /S switch, 4-4, 4-11
FILCOM source file spec,
4-1
FILCOM switches, 4-2, 4-11
FILCOM /U switch, 4-4, 4-11
FILCOM /U switch example,
4-7
FILCOM /W switch, 4-8, 4-11
FILCOM /W switch example,
4-10
FILCOM /X switch, 4-8, 4-11
File,
COMND.CMD, 3-2
CREF indirect, 5-5

INDEX (CONT.)

File (Cont.)
 CREF SWITCH.INI, 5-5
 .EXE, 6-1
 MAIL indirect, 2-4
 MAIL.CPY, 2-3, 2-9
 MAIL.TXT, 2-9, 3-1
 MAKLIB indirect, 6-3
 MAKLIB SWITCH.INI, 6-3
 Overwriting a disk, 7-10,
 7-27
 .REL, 6-1
 File access errors,
 CREF, 5-13
 FILCOM, 4-13, 4-14
 File account during RESTORE,
 Specifying, 7-14
 File comparison,
 ASCII mode, 4-3
 Blank lines in, 4-3
 Comments and spaces in,
 4-3
 Limited, 4-8
 Match criteria in, 4-4
 Nonsharable save, 4-8
 Nonstandard, 4-8
 Quick, 4-4, 4-8
 Sharable save, 4-8
 Spaces and tabs in, 4-4
 status of, 4-3
 Update mode, 4-4
 File comparison switches,
 ASCII, 4-3
 Binary, 4-8
 File Descriptor Blocks,
 DUMPER, 7-28
 File format,
 ASCII output, 4-4
 Binary output, 4-8
 File generation numbers,
 1-2
 File header,
 ASCII output, 4-4
 Binary output, 4-8
 File of filenames during
 SAVE,
 Producing a, 7-12
 File of user names, 2-4
 File spec,
 CREF destination, 5-3
 CREF source, 5-3
 DUMPER destination, 7-23
 DUMPER source, 7-23
 FILCOM destination, 4-1
 FILCOM source, 4-1
 MAKLIB destination, 6-3
 MAKLIB source, 6-3
 File specs,
 Fields of, 1-2
 Typing, 1-1, 1-2
 File status,
 Invisible, 7-48
 ;OFFLINE, 7-49
 File types,
 Binary, 4-2
 FILCOM, 4-2
 FILCOM nonstandard, 4-8,
 4-11
 Nonsharable save, 4-2
 Offset address, 4-2
 Sharable save, 4-2
 Filenames during SAVE,
 Producing a file of, 7-12
 Files,
 ALGOL compiler source,
 5-1
 Archived, 7-1
 Archiving, 7-3, 7-43,
 7-47, 7-48
 Archiving marked, 7-46
 Backing up, 7-44
 Backup, 7-1
 COBOL compiler source,
 5-1
 Comparing ASCII, 4-3
 Comparing assembly
 language source, 4-3,
 4-11
 Comparing binary, 4-7
 Creating .CRF, 5-1
 Creating .FIX, 6-20
 Dumping, 4-8
 FORTRAN compiler source,
 5-1
 Listing binary, 4-7
 MACRO assembler source,
 5-1
 MACRO ASSEMBLER source,
 4-11
 MACRO Assembler source,
 4-3
 Master library, 6-7
 Migrated, 7-1
 Migrating, 7-3, 7-43,
 7-49
 Nonsharable save, 4-2,
 4-7, 4-8, 4-11
 relocatable, 6-1
 Restoring archived and
 migrated, 7-50
 Retrieving archived and
 migrated, 7-50
 Sharable save, 4-2, 4-7,
 4-8, 4-11
 Transaction, 6-3
 FILES command,
 DUMPER, 7-8, 7-11, 7-52
 Files during RESTORE,
 Renaming, 7-26

INDEX (CONT.)

- Files during SAVE,
 - Renaming, 7-24
- Files for archiving,
 - Marking, 7-31
- Files from backup,
 - Restoring, 7-46
- Files from disk to tape,
 - Transferring, 7-24
- Files from tape to disk,
 - Transferring, 7-26
- Files incrementally,
 - Saving, 7-43
- .FIX files,
 - Creating, 6-20
- .FIX files code format,
 - 6-43
- .FIX files example, 6-26
- /FIX switch,
 - MAKLIB, 6-5, 6-25, 6-26, 6-27
- /FIX switch example,
 - MAKLIB, 6-26
- FORMAT command,
 - DUMPER, 7-8, 7-16, 7-53
- FORTTRAN compiler source files, 5-1
- FORTTRAN CREF listing example, 5-8
- /FULL-INCREMENTAL switch,
 - DUMPER, 7-42, 7-44, 7-55

- Generating an .EXE file
 - figure, 6-1
- Generating libraries figure,
 - 6-2
- Generation numbers,
 - File, 1-2
- Generation numbers in
 - FILCOM, 4-2
- GET command, 2-3
- GET command example, 2-3
- Global symbols, 6-20

- /H switch,
 - CREF, 5-4
 - FILCOM, 4-3, 4-8, 4-11
- Halting a command in DUMPER,
 - 7-29
- Header,
 - ASCII output file, 4-4
 - Binary output file, 4-8
- HELP command,
 - DUMPER, 7-7, 7-53
- /HELP switch,
 - RDMAIL, 3-4

- Help text,
 - FILCOM, 4-3, 4-8, 4-11
 - RDMAIL, 3-4
- High-segment break, 6-4

- Identifying a mounted tape,
 - 7-14
- Inactive edit, 6-20
- /INCREMENTAL:n switch,
 - DUMPER, 7-42, 7-44, 7-55
- INDEX blocks,
 - Inserting, 6-19
- /INDEX switch,
 - MAKLIB, 6-19, 6-27
- /INDEX switch example,
 - MAKLIB, 6-20
- Indirect file,
 - CREF, 5-5
 - MAIL, 2-4
 - MAKLIB, 6-3
- Indirect file example,
 - MAIL, 2-4
- Indirect file in MAIL,
 - Retrieving an, 2-4
- INDUSTRY command,
 - DUMPER, 7-8, 7-16, 7-53
- Information about libraries,
 - Obtaining, 6-2, 6-4
- INFORMATION ARCHIVE-STATUS
 - command, 7-31
- INFORMATION ARCHIVE-STATUS
 - command example, 7-47
- INFORMATION MAIL command,
 - 2-4, 3-2
- INFORMATION SYSTEM-STATUS
 - command, 7-3
- INFORMATION SYSTEM-STATUS
 - command example, 7-3
- INFORMATION VOLUMES command,
 - 7-6
- INFORMATION VOLUMES command
 - example, 7-6
- Informational messages,
 - CREF, 5-10
 - PLEASE, 8-6
- INITIAL command,
 - DUMPER, 7-8, 7-19, 7-53
- Initials,
 - Edit creator, 6-21
- Initials MAKLIB user, 6-26
- .INSERT pseudo-op, 6-22,
 - 6-23
- .INSERT pseudo-op example,
 - 6-23
- /INSERT switch,
 - MAKLIB, 6-7, 6-13, 6-27

INDEX (CONT.)

- /INSERT switch example,
MAKLIB, 6-15, 6-17
- /INSERT switch figure,
MAKLIB, 6-14, 6-15
- Inserting code in modules,
6-22
- Inserting INDEX blocks,
6-19
- Inserting modules into
libraries, 6-13
- Inter-Process Communication
Facility, 2-9
- INTERCHANGE command,
DUMPER, 7-8, 7-16, 7-53
- INTERCHANGE mode, 7-15
- Invalid RDMAIL switches,
3-7
- Invalid user names, 2-1
- Invisible file status, 7-48
- Invoking the utilities,
1-1
- IPCF packet, 2-9

- JFN, 7-2
- Job file number, 7-2

- /K switch,
CREF, 5-4

- Labeled tape, 7-3, 7-5
- Libraries, 6-1
 - Appending modules to, 6-7
 - Creating, 6-9
 - Editing, 6-2, 6-20
 - Extracting modules from,
6-11
 - Inserting modules into,
6-13
 - Listing edits to, 6-5
 - Listing entry points in,
6-5
 - Listing modules in, 6-4
 - Loading instructions in,
6-6
 - Manipulating, 6-2, 6-7
 - Master, 6-3
 - Modifying, 6-2, 6-19
 - Obtaining information
about, 6-2, 6-4
 - Removing modules from,
6-9
 - Replacing modules in,
6-17
- Libraries figure,
Generating, 6-2
- Limited file comparison,
4-8
- Line printer,
Printing your messages on
the, 3-5
- LINK program, 6-1, 6-2, 6-5,
6-19
- LIST command,
DUMPER, 7-8, 7-12, 7-53
- /LIST switch,
MAKLIB, 6-4, 6-27
- RDMAIL, 3-4, 3-5
- /LIST switch example,
MAKLIB, 6-9, 6-11, 6-13,
6-15, 6-17, 6-19
- Listing binary files, 4-7
- Listing edits to libraries,
6-5
- Listing entry points in
libraries, 6-5
- Listing modules in
libraries, 6-4
- Listings,
Cross-reference, 5-1
- Producing cross-reference,
5-2
- tables in cross-reference,
5-2
- /LOAD switch,
MAKLIB, 6-4, 6-6, 6-27
- /LOAD switch example,
MAKLIB, 6-6
- LOAD-class command, 5-1
- Loading instructions in
libraries, 6-6
- Local symbols,
Deleting, 6-20
- Logical names, 1-2, 1-3
- Logical names example,
Defining, 1-3, 5-6, 7-4,
7-5
- Logical names in DUMPER, 7-5
- Logical names in FILCOM,
4-2
- Logical records,
Setting, 7-15
- Low-segment break, 6-4

- /M switch,
CREF, 5-4
- MACRO assembler,
MAKLIB, 6-24, 6-25
- MACRO assembler source
files, 5-1

INDEX (CONT.)

- MACRO ASSEMBLER source
 - files, 4-11
- MACRO Assembler source
 - files, 4-3
- MACRO CREF listing example,
 - 5-6, 5-21
- Macro names table,
 - OPDEF and, 5-2
- MAIL,
 - Enabling capabilities in,
 - 2-5
- Mail,
 - Errors in sending, 2-3
 - notification of new, 2-2,
 - 3-1, 3-2
 - Reading another user's,
 - 3-2
 - Reading new, 3-3
 - Resending, 2-3
- MAIL,
 - Retrieving an indirect
 - file in, 2-4
 - Running, 2-1
- Mail,
 - status of new, 2-4, 3-2
- MAIL CTRL/H, 2-1, 2-4
- MAIL CTRL/V, 2-2
- MAIL CTRL/Z, 2-2
- MAIL error messages, 2-3,
 - 2-6
- MAIL error recovery
 - procedure, 2-3
- MAIL indirect file, 2-4
- MAIL indirect file example,
 - 2-4
- MAIL message,
 - Terminating a, 2-2
- MAIL multi-line entries, 2-2
- MAIL options, 2-4
- MAIL program, 2-1
- MAIL recipients, 2-1
- MAIL subject line, 2-2
- MAIL subject line special
 - characters, 2-2
- MAIL technical notes, 2-9
- MAIL used in a batch job,
 - 2-6
- MAIL warning messages, 2-6
- MAIL.CPY file, 2-3, 2-9
- MAIL.TXT file, 2-9, 3-1
- MAIL/MAILER figure, 2-9
- MAILER program, 2-9
- MAKLIB,
 - Running, 6-3
 - stopping, 6-4
- MAKLIB /APPEND switch, 6-7,
 - 6-27
- MAKLIB /APPEND switch
 - example, 6-9
- MAKLIB /APPEND switch
 - figure, 6-8
- MAKLIB code ending point,
 - 6-23
- MAKLIB command string
 - format, 6-3
- MAKLIB command string
 - requirements, 6-4, 6-7,
 - 6-19, 6-25
- MAKLIB /DELETE switch, 6-7,
 - 6-9, 6-20, 6-27
- MAKLIB /DELETE switch
 - example, 6-11
- MAKLIB /DELETE switch
 - figure, 6-10
- MAKLIB destination file
 - spec, 6-3
- MAKLIB edit ending point,
 - 6-23
- MAKLIB edit example, 6-23,
 - 6-26
- MAKLIB error messages, 6-28
- MAKLIB /EXIT switch, 6-27
- MAKLIB /EXTRACT switch, 6-7,
 - 6-11, 6-20, 6-27
- MAKLIB /EXTRACT switch
 - example, 6-13
- MAKLIB /EXTRACT switch
 - figure, 6-11
- MAKLIB /FIX switch, 6-5,
 - 6-25, 6-26, 6-27
- MAKLIB /FIX switch example,
 - 6-26
- MAKLIB functions, 6-2
- MAKLIB /INDEX switch, 6-19,
 - 6-27
- MAKLIB /INDEX switch
 - example, 6-20
- MAKLIB indirect file, 6-3
- MAKLIB /INSERT switch, 6-7,
 - 6-13, 6-27
- MAKLIB /INSERT switch
 - example, 6-15, 6-17
- MAKLIB /INSERT switch
 - figure, 6-14, 6-15
- MAKLIB /LIST switch, 6-4,
 - 6-27
- MAKLIB /LIST switch example,
 - 6-4, 6-9, 6-11, 6-13,
 - 6-15, 6-17, 6-19
- MAKLIB /LOAD switch, 6-4,
 - 6-6, 6-27
- MAKLIB /LOAD switch example,
 - 6-6
- MAKLIB MACRO assembler,
 - 6-24, 6-25
- MAKLIB /MASTER switch, 6-7,
 - 6-27

INDEX (CONT.)

- MAKLIB /MASTER switch
 - example, 6-15, 6-17, 6-18
- MAKLIB message variables, 6-28
- MAKLIB multi-line commands, 6-4
- MAKLIB /NOLOCALS switch, 6-19, 6-20, 6-27
- MAKLIB /NOLOCALS switch
 - example, 6-20
- MAKLIB /POINTS switch, 6-4, 6-5, 6-27
- MAKLIB /POINTS switch
 - example, 6-5
- MAKLIB program, 6-1
- MAKLIB /REPLACE switch, 6-7, 6-17, 6-20, 6-27
- MAKLIB /REPLACE switch
 - example, 6-18
- MAKLIB /REPLACE switch
 - figure, 6-17
- MAKLIB source file spec, 6-3
- MAKLIB switch arguments, 6-3
- MAKLIB SWITCH.INI file, 6-3
- MAKLIB switches, 6-2, 6-3
- MAKLIB technical notes, 6-41
- MAKLIB /TRACE switch, 6-4, 6-5, 6-20, 6-27
- MAKLIB /TRACE switch
 - example, 6-6
- MAKLIB user,
 - Initials, 6-26
- MAKLIB warning messages, 6-28
- MAKLIB /WHO switch, 6-6, 6-25, 6-26, 6-27
- MAKLIB /WHO switch example, 6-26
- Manipulating libraries, 6-2, 6-7
- Marked files,
 - Archiving, 7-46
- Marking files for archiving, 7-31
- Master libraries, 6-3
- Master library files, 6-7
- /MASTER switch,
 - MAKLIB, 6-7, 6-27
- /MASTER switch example,
 - MAKLIB, 6-15, 6-17, 6-18
- Match criteria in file
 - comparison, 4-4
- MBEFORE command,
 - DUMPER, 7-8, 7-9, 7-53
- MESSAGE command,
 - PLEASE, 8-5
- MESSAGE command line format,
 - PLEASE, 8-5
- /MESSAGE switch,
 - PLEASE, 8-3
- Message variables,
 - CREP, 5-10
 - DUMPER, 7-57
 - FILCOM, 4-12
 - MAKLIB, 6-28
- Message-of-the-Day, 2-5, 3-1
- Message-of-the-Day,
 - Reading a, 3-5
- /MESSAGE-OF-THE-DAY switch,
 - RDMAIL, 3-4, 3-5
- Messages,
 - CREP error, 5-10
 - CREP informational, 5-10
 - CREP warning, 5-10
 - DUMPER error, 7-57
 - DUMPER warning, 7-57
 - FILCOM error, 4-12
 - MAIL error, 2-3, 2-6
 - MAIL warning, 2-6
 - MAKLIB error, 6-28
 - MAKLIB warning, 6-28
 - Perusing your, 3-6
 - PLEASE error, 8-6
 - PLEASE informational, 8-6
 - PLEASE warning, 8-6
 - RDMAIL error, 3-7
 - RDMAIL warning, 3-7
 - Reading all your, 3-5
 - Stopping your, 3-6
- Messages on the line
 - printer,
 - Printing your, 3-5
- /MIGRATE switch,
 - DUMPER, 7-43, 7-49, 7-55
- Migrated files, 7-1
- Restoring archived and, 7-50
- Retrieving archived and, 7-50
- Migrating files, 7-3, 7-43, 7-49
- Modifying libraries, 6-2, 6-19
- .MODULE pseudo-op, 6-21
- Modules,
 - Code and symbol table in, 6-25
 - Deleting code in, 6-22
 - Inserting code in, 6-22
 - REL, 6-1
 - Relocatable object, 6-1

INDEX (CONT.)

Modules in libraries,
 Listing, 6-4
 MOUNT TAPE command,
 DUMPER, 7-4, 7-6
 Mounted tape,
 Identifying a, 7-14
 Mounting a tape, 7-4
 MSINCE command,
 DUMPER, 7-8, 7-9, 7-53
 MTA-DUMPER:, 7-4, 7-5, 7-14
 Multi-line commands,
 MAKLIB, 6-4
 Multi-line entries,
 MAIL, 2-2
 PLEASE, 8-2
 Multi-volume tape set, 7-5

 .NAME pseudo-op, 6-6, 6-21
 New mail,
 notification of, 2-2, 3-1,
 3-2
 Reading, 3-3
 status of, 2-4, 3-2
 /nL switch,
 FILCOM, 4-4, 4-8, 4-11
 NO CHECKSUM command,
 DUMPER, 7-51
 NO CREATE command,
 DUMPER, 7-52
 NO DIRECTORIES command,
 DUMPER, 7-11, 7-52
 NO FILES command,
 DUMPER, 7-12, 7-52
 NO INDUSTRY command,
 DUMPER, 7-53
 NO INTERCHANGE command,
 DUMPER, 7-53
 NO LIST command,
 DUMPER, 7-13, 7-53
 NO SILENCE command,
 DUMPER, 7-13, 7-55
 Node names, 8-3
 /NOINCREMENTAL switch,
 DUMPER, 7-43, 7-44, 7-55
 /NOLOCALS switch,
 MAKLIB, 6-19, 6-20, 6-27
 /NOLOCALS switch example,
 MAKLIB, 6-20
 Non-files-only directory,
 Sending mail to a, 2-5
 Nonprivileged user in
 DUMPER, 7-7
 Nonsharable save file
 comparison, 4-8
 Nonsharable save file types,
 4-2

 Nonsharable save files, 4-2,
 4-7, 4-8, 4-11
 Nonstandard file comparison,
 4-8
 Nonstandard file types,
 FILCOM, 4-8, 4-11
 /NOUNLOAD switch,
 DUMPER, 7-22
 /NOWAIT switch,
 PLEASE, 8-3
 /nU switch,
 FILCOM, 4-8, 4-11

 /O switch,
 CREF, 5-4
 Obtaining information about
 libraries, 6-2, 6-4
 Octal codes,
 CREF, 5-13
 FILCOM, 4-13, 4-14
 ;OFFLINE file status, 7-49
 Offset address file types,
 4-2
 One-segment program, 6-4
 Opcode table, 5-2
 Opcode table example, 5-6
 OPDEF and Macro names table,
 5-2
 Operator,
 Communicating with the,
 8-1
 OPERATOR capabilities, 2-5,
 3-2, 7-1
 Output file format,
 ASCII, 4-4
 Binary, 4-8
 Output file header,
 ASCII, 4-4
 Binary, 4-8
 Outstanding PLEASE messages,
 Canceling, 8-5
 Overwriting a disk file,
 7-10, 7-27

 /P switch,
 CREF, 5-4
 Parity,
 Setting tape, 7-15
 PARITY command,
 DUMPER, 7-8, 7-15, 7-53
 Period to replace user
 names, 2-6
 /PERUSE switch,
 RDMAIL, 3-4, 3-6

INDEX (CONT.)

Perusing your messages, 3-6
 PLEASE,
 Running, 8-1
 Stopping, 8-2, 8-5
 PLEASE at PLEASE program
 level, 8-1, 8-4
 PLEASE at TOPS-20 command
 level, 8-1, 8-2
 PLEASE CANCEL command, 8-5
 PLEASE CANCEL command
 example, 8-5
 PLEASE command,
 Batch job, 8-1
 PLEASE commands, 8-1, 8-4
 PLEASE error messages, 8-6
 PLEASE EXIT command, 8-5
 PLEASE EXIT command example,
 8-5
 PLEASE informational
 messages, 8-6
 PLEASE MESSAGE command, 8-5
 PLEASE MESSAGE command
 example, 8-5
 PLEASE MESSAGE command line
 format, 8-5
 PLEASE /MESSAGE switch, 8-3
 PLEASE /MESSAGE switch
 example, 8-3
 PLEASE message to a node,
 Sending a, 8-3
 PLEASE messages,
 Canceling outstanding,
 8-5
 PLEASE multi-line entries,
 8-2
 PLEASE /NOWAIT switch, 8-3
 PLEASE /NOWAIT switch
 example, 8-3
 PLEASE operator response,
 Saving a, 8-2
 PLEASE program, 8-1
 PLEASE program level,
 PLEASE at, 8-1, 8-4
 PLEASE program level
 example, 8-4
 PLEASE SEND command, 8-4
 PLEASE SEND command example,
 8-4
 PLEASE SEND command line
 format, 8-4
 PLEASE special characters,
 8-2
 PLEASE switches, 8-3
 PLEASE TOPS-20 command
 level example, 8-2
 PLEASE warning messages, 8-6
 /POINTS switch,
 MAKLIB, 6-4, 6-5, 6-27
 /POINTS switch example,
 MAKLIB, 6-5
 Positioning at beginning of
 tape, 7-21
 Positioning at End-of-Tape,
 7-21
 Positioning tape, 7-20
 PPN, 1-2, 5-3
 Preserving a saveset, 7-26
 PRINT command,
 DUMPER, 7-29, 7-54
 Printing directory names
 during SAVE or RESTORE,
 7-11
 Printing filenames during
 SAVE and RESTORE, 7-3
 Printing filenames during
 SAVE or RESTORE, 7-11
 Printing your messages on
 the line printer, 3-5
 Privileged user in DUMPER,
 7-42
 Procedure nesting table,
 5-2
 Procedure nesting table
 example, 5-9
 Producing a file of
 filenames during SAVE,
 7-12
 Producing cross-reference
 listings, 5-2
 Program,
 Compiling a, 5-1, 5-2
 CREF, 5-1
 DUMPER, 7-1
 Executable, 6-1
 FILCOM, 4-1
 LINK, 6-1, 6-2, 6-5, 6-19
 MAIL, 2-1
 MAILER, 2-9
 MAKLIB, 6-1
 One-segment, 6-4
 PLEASE, 8-1
 RDMAIL, 3-1
 Running CREF as a, 5-5
 SCAN, 6-28
 Two-segment, 6-4
 WILD, 6-28
 Program break, 6-4
 Project-programmer number,
 1-2, 5-3
 Protection code during
 RESTORE,
 Specifying, 7-14
 PROTECTION command,
 DUMPER, 7-8, 7-14, 7-54
 PS:<REMARKS> directory,
 2-5

INDEX (CONT.)

PS:<REMARKS> example,
 Sending mail to, 2-5
 PS:<SYSTEM> directory, 2-5,
 2-6
 PSECTS, 6-20
 Pseudo-op,
 .ALTER, 6-21
 .ASSOCIATED, 6-21
 Control, 6-20
 .DATE, 6-6, 6-21
 .EDIT, 6-20, 6-21
 .ENDE, 6-20, 6-23
 .ENDI, 6-22, 6-23
 .INSERT, 6-22, 6-23
 .MODULE, 6-21
 .NAME, 6-6, 6-21
 .REINSERT, 6-6, 6-23,
 6-26
 .REMOVE, 6-6, 6-23,
 6-26
 .VERSION, 6-21
 Pseudo-op order figure,
 6-24
 Pseudo-ops, 6-21

 /Q switch,
 FILCOM, 4-4, 4-8, 4-11
 Quick file comparison, 4-4,
 4-8
 QUIT command,
 DUMPER, 7-7, 7-54

 /R switch,
 CREF, 5-4
 Radix-50 characters, 6-21
 RDMAIL,
 Enabling capabilities in,
 3-2
 Running, 3-2
 RDMAIL /ALL switch, 3-4,
 3-5
 RDMAIL /ALL switch example,
 3-5
 RDMAIL date and time, 3-3
 RDMAIL error messages, 3-7
 RDMAIL /HELP switch, 3-4
 RDMAIL /HELP switch example,
 3-4
 RDMAIL help text, 3-4
 RDMAIL /LIST switch, 3-4,
 3-5
 RDMAIL /LIST switch example,
 3-5
 RDMAIL /MESSAGE-OF-THE-DAY
 switch, 3-4, 3-5

 RDMAIL /MESSAGE-OF-THE-DAY
 switch example, 3-5
 RDMAIL /PERUSE switch, 3-4,
 3-6
 RDMAIL /PERUSE switch
 example, 3-6
 RDMAIL program, 3-1
 RDMAIL /STOP switch, 3-4,
 3-6
 RDMAIL /STOP switch example,
 3-6
 RDMAIL switches, 3-3
 Invalid, 3-7
 using, 3-4
 RDMAIL warning messages,
 3-7
 Reading a
 Message-of-the-Day, 3-5
 Reading a tape from an
 earlier version of
 DUMPER, 7-16
 Reading a TOPS-10 tape,
 Writing and, 7-16
 Reading all your messages,
 3-5
 Reading and writing tape in
 36 bits, 7-16
 Reading another user's mail,
 3-2
 Reading another user's mail
 example, 3-2
 Reading new mail, 3-3
 Recognition, 1-2
 Record numbers,
 Tape, 7-26
 Recording filenames from a
 tape, 7-29
 Recovering from a system
 crash, 7-2, 7-19, 7-37,
 7-45
 Recovery procedure,
 MAIL error, 2-3
 REENTER command, 2-3
 REENTER command example,
 2-3
 REFUSE SYSTEM-MESSAGES
 command, 3-2
 Regular symbols table, 5-2
 .REINSERT pseudo-op, 6-6,
 6-23, 6-26
 .REL file, 6-1
 REL modules, 6-1
 Related edit, 6-21
 Releasing a tape, 7-23
 Relocatable files, 6-1
 Relocatable object modules,
 6-1
 .REMOVE pseudo-op, 6-6,
 6-23, 6-26

INDEX (CONT.)

- Removing modules from libraries, 6-9
- Renaming files during RESTORE, 7-26
- Renaming files during SAVE, 7-24
- REPEAT LOGIN-MESSAGES subcommand, 3-1
- /REPLACE switch, MAKLIB, 6-7, 6-17, 6-20, 6-27
- /REPLACE switch example, MAKLIB, 6-18
- /REPLACE switch figure, MAKLIB, 6-17
- Replacing modules in libraries, 6-17
- REQUEST blocks, 6-6
- REQUIRE blocks, 6-6
- Resending mail, 2-3
- RESET command, 8-2
- RESTORE,
 - Printing directory names during SAVE or, 7-11
 - Printing filenames during SAVE and, 7-3
 - Printing filenames during SAVE or, 7-11
 - Renaming files during, 7-26
 - Specifying file account during, 7-14
 - Specifying protection code during, 7-14
 - Supressing directory names and filenames during SAVE or, 7-13
- RESTORE command, DUMPER, 7-23, 7-26, 7-43, 7-46, 7-54
- Restoring,
 - Continuing, 7-19
- Restoring a directory, 7-46
- Restoring archived and migrated files, 7-50
- Restoring files from backup, 7-46
- Restriction,
 - Code insertion, 6-22
- RETRIEVE command, DUMPER, 7-43, 7-50, 7-54
- Retrieving an indirect file in MAIL, 2-4
- Retrieving archived and migrated files, 7-50
- Retrieving user names, 2-1
- REWIND command, DUMPER, 7-20, 7-21, 7-54
- REWIND SWITCHING command, DUMPER, 7-21
- Running CREF, 5-1
- Running CREF as a command, 5-5
- Running CREF as a command example, 5-5
- Running CREF as a program, 5-5
- Running CREF as a program example, 5-5
- Running DUMPER, 7-6
- Running FILCOM, 4-1
- Running MAIL, 2-1
- Running MAKLIB, 6-3
- Running other programs from CREF, 5-6
- Running PLEASE, 8-1
- Running RDMAIL, 3-2
- /S switch,
 - CREF, 5-4
 - FILCOM, 4-4, 4-11
- SAVE,
 - Producing a file of filenames during, 7-12
 - Renaming files during, 7-24
 - Specifying a checksum during, 7-18
- SAVE and RESTORE,
 - Printing filenames during, 7-3
- SAVE command, DUMPER, 7-23, 7-24, 7-55
- SAVE or RESTORE,
 - Printing directory names during, 7-11
 - Printing filenames during, 7-11
 - Supressing directory names and filenames during, 7-13
- Saveset, 7-2
 - Creating a, 7-24
 - Deleting a, 7-25
 - Preserving a, 7-26
 - Skipping a, 7-22
- Saveset name, 7-2
 - Specifying a, 7-18
- Saving,
 - Continuing, 7-19
- Saving a PLEASE operator response, 8-2
- Saving files incrementally, 7-43

INDEX (CONT.)

Saving files on multiple
 tape, 7-2
 SCAN program, 6-28
 SEND command,
 PLEASE, 8-4
 SEND command line format,
 PLEASE, 8-4
 Sending a PLEASE message to
 a node, 8-3
 Sending mail,
 Errors in, 2-3
 Sending mail to a
 non-files-only
 directory, 2-5
 Sending mail to
 PS:<REMARKS> example,
 2-5
 Sending mail to PS:<SYSTEM>
 example, 2-5
 Set,
 Switching between volumes
 in a, 7-21
 SET BLOCKING-FACTOR command,
 DUMPER, 7-8, 7-15, 7-16,
 7-55
 SET MAIL-WATCH command, 3-2
 SET TAPE-NUMBER command,
 DUMPER, 7-8, 7-19, 7-55
 Setting logical records,
 7-15
 Setting tape density, 7-15
 Setting tape parity, 7-15
 Sharable save file
 comparison, 4-8
 Sharable save file types,
 4-2
 Sharable save files, 4-2,
 4-7, 4-8, 4-11
 SILENCE command,
 DUMPER, 7-8, 7-13, 7-55
 SINCE command,
 DUMPER, 7-8, 7-9, 7-55
 SIXBIT characters, 6-21
 SKIP command,
 DUMPER, 7-22, 7-56
 SKIP n command,
 DUMPER, 7-20
 Skipping a saveset, 7-22
 Source file spec,
 CREF, 5-3
 DUMPER, 7-23
 FILCOM, 4-1
 MAKLIB, 6-3
 Source files,
 ALGOL compiler, 5-1
 COBOL compiler, 5-1
 Comparing assembly
 language, 4-3, 4-11
 FORTRAN compiler, 5-1

Source files (Cont.)
 MACRO assembler, 5-1
 MACRO ASSEMBLER, 4-11
 MACRO Assembler, 4-3
 Spaces and tabs in file
 comparison, 4-4
 Spaces in file comparison,
 Comments and, 4-3
 Special characters,
 MAIL subject line, 2-2
 PLEASE, 8-2
 Specifying a checksum
 during SAVE, 7-18
 Specifying a saveset name,
 7-18
 Specifying file account
 during RESTORE, 7-14
 Specifying protection code
 during RESTORE, 7-14
 SSNAME command,
 DUMPER, 7-8, 7-18, 7-56
 Status-Setting commands,
 DUMPER, 7-7, 7-8
 /STOP switch,
 RDMAIL, 3-4, 3-6
 Stopping DUMPER, 7-7
 Stopping FILCOM, 4-3
 Stopping MAKLIB, 6-4
 Stopping PLEASE, 8-2, 8-5
 Stopping your messages,
 3-6
 Subcommand,
 REPEAT LOGIN-MESSAGES,
 3-1
 Subject line,
 MAIL, 2-2
 SUPERSEDE command,
 DUMPER, 7-8, 7-10, 7-56
 Suppressing directory names
 and filenames during
 SAVE or RESTORE, 7-13
 Switch,
 /CREF, 5-1, 5-2
 CREF /A, 5-4
 CREF /B, 5-4
 CREF /C, 5-4
 CREF /D, 5-4
 CREF /H, 5-4
 CREF /K, 5-4
 CREF /M, 5-4
 CREF /O, 5-4
 CREF /P, 5-4
 CREF /R, 5-4
 CREF /S, 5-4
 CREF /T, 5-4
 CREF /W, 5-4
 CREF /Z, 5-4
 DUMPER /ARCHIVE, 7-43,
 7-48, 7-55

INDEX (CONT.)

Switch (Cont.)

DUMPER /FULL-INCREMENTAL,
 7-42, 7-44, 7-55
 DUMPER /INCREMENTAL:n,
 7-42, 7-44, 7-55
 DUMPER /MIGRATE, 7-43,
 7-49, 7-55
 DUMPER /NOINCREMENTAL,
 7-43, 7-44, 7-55
 DUMPER /NOUNLOAD, 7-22
 DUMPER /TAPE-INFORMATION,
 7-51
 DUMPER /VOLIDS, 7-5
 DUMPER /WRITE-ENABLED,
 7-4
 FILCOM /A, 4-3, 4-11
 FILCOM /B, 4-3, 4-11
 FILCOM /C, 4-3, 4-11
 FILCOM /E, 4-8, 4-11
 FILCOM /H, 4-3, 4-8, 4-11
 FILCOM /nL, 4-4, 4-8,
 4-11
 FILCOM /nU, 4-8, 4-11
 FILCOM /Q, 4-4, 4-8,
 4-11
 FILCOM /S, 4-4, 4-11
 FILCOM /U, 4-4, 4-11
 FILCOM /W, 4-8, 4-11
 FILCOM /X, 4-8, 4-11
 MAKLIB /APPEND, 6-7, 6-27
 MAKLIB /DELETE, 6-7, 6-9,
 6-20, 6-27
 MAKLIB /EXIT, 6-27
 MAKLIB /EXTRACT, 6-7,
 6-11, 6-20, 6-27
 MAKLIB /FIX, 6-5, 6-25,
 6-26, 6-27
 MAKLIB /INDEX, 6-19, 6-27
 MAKLIB /INSERT, 6-7, 6-13,
 6-27
 MAKLIB /LIST, 6-4, 6-27
 MAKLIB /LOAD, 6-4, 6-6,
 6-27
 MAKLIB /MASTER, 6-7, 6-27
 MAKLIB /NOLOCALS, 6-19,
 6-20, 6-27
 MAKLIB /POINTS, 6-4, 6-5,
 6-27
 MAKLIB /REPLACE, 6-7,
 6-17, 6-20, 6-27
 MAKLIB /TRACE, 6-4, 6-5,
 6-20, 6-27
 MAKLIB /WHO, 6-6, 6-25,
 6-26, 6-27
 PLEASE /MESSAGE, 8-3
 PLEASE /NOWAIT, 8-3
 RDMAIL /ALL, 3-4, 3-5
 RDMAIL /HELP, 3-4
 RDMAIL /LIST, 3-4, 3-5

Switch (Cont.)

RDMAIL
 /MESSAGE-OF-THE-DAY,
 3-4, 3-5
 RDMAIL /PERUSE, 3-4, 3-6
 RDMAIL /STOP, 3-4, 3-6
 Switch arguments,
 MAKLIB, 6-3
 SWITCH.INI file,
 CREF, 5-5
 MAKLIB, 6-3
 SWITCH.INI file example,
 5-6
 Switches,
 ASCII file comparison,
 4-3
 Binary file comparison,
 4-8
 CREF, 5-3
 FILCOM, 4-2, 4-11
 Invalid RDMAIL, 3-7
 MAKLIB, 6-2, 6-3
 PLEASE, 8-3
 RDMAIL, 3-3
 using RDMAIL, 3-4
 Switching between volumes
 in a set, 7-21
 SYMBOL blocks, 6-20
 Symbols,
 Deleting local, 6-20
 Global, 6-20
 System crash,
 Recovering from a, 7-2,
 7-19, 7-37, 7-45

 /T switch,
 CREF, 5-4
 Table,
 Opcode, 5-2
 OPDEF and Macro names, 5-2
 Procedure nesting, 5-2
 Regular symbols, 5-2
 Tabs in file comparison,
 Spaces and, 4-4
 TALK command, 2-4
 Tape,
 Dismounting a, 7-5
 Identifying a mounted,
 7-14
 Labeled, 7-3, 7-5
 Mounting a, 7-4
 Positioning, 7-20
 Recording filenames from
 a, 7-29
 Releasing a, 7-23
 Saving files on multiple,
 7-2

INDEX (CONT.)

- Tape (Cont.)
 - Transferring between disk and, 7-2
 - Transferring files from disk to, 7-24
 - Unlabeled, 7-3, 7-5
- TAPE command,
 - DUMPER, 7-8, 7-14, 7-56
- Tape density,
 - Setting, 7-15
- Tape drive,
 - Assigning a, 7-4
- Tape drive allocation, 7-3
 - status of, 7-3
- Tape drive example,
 - Assigning a, 7-4
- Tape from an earlier version of DUMPER,
 - Reading a, 7-16
- Tape parity,
 - Setting, 7-15
- Tape record numbers, 7-26
- Tape set, 7-2
 - Multi-volume, 7-5
 - Verifying another tape in a, 7-6
- Tape to disk,
 - Transferring files from, 7-26
- /TAPE-INFORMATION switch,
 - DUMPER, 7-51
- Tape-Positioning commands,
 - DUMPER, 7-7, 7-20
- Technical notes,
 - CREF, 5-14
 - MAIL, 2-9
 - MAKLIB, 6-41
- Terminating a MAIL message, 2-2
- TOPS-10 tape,
 - Writing and reading a, 7-16
- TOPS-20 command level,
 - PLEASE at, 8-1, 8-2
- TRACE blocks, 6-5, 6-20, 6-21, 6-26
- TRACE blocks format, 6-41
- /TRACE switch,
 - MAKLIB, 6-4, 6-5, 6-20, 6-27
- /TRACE switch example,
 - MAKLIB, 6-6
- Transaction files, 6-3
- Transfer commands,
 - DUMPER, 7-23
- Transferring between disk and tape, 7-2
- Transferring files by date and time, 7-8, 7-9
- Transferring files from disk to tape, 7-24
- Transferring files from tape to disk, 7-26
- TRANSLATE command, 1-2
- TRANSLATE command example, 1-2
- Two-segment program, 6-4
- TYPE command, 4-5, 5-6, 6-5
- TYPE command example, 4-5, 4-6, 5-6, 6-5
- Typing file specs, 1-1, 1-2
- /U switch,
 - FILCOM, 4-4, 4-11
- Unlabeled tape, 7-3, 7-5
- UNLOAD command,
 - DUMPER, 7-20, 7-23, 7-56
- Update mode file comparison, 4-4
- User names,
 - Duplicate, 2-1
 - File of, 2-4
 - Invalid, 2-1
 - Period to replace, 2-6
 - Retrieving, 2-1
- Utilities,
 - Invoking the, 1-1
- Variables,
 - CREF message, 5-10
 - DUMPER message, 7-57
 - FILCOM message, 4-12
 - MAKLIB message, 6-28
- VDIRECTORY command example, 7-10
- Verifying another tape in a tape set, 7-6
- Version number,
 - Edit, 6-21
- .VERSION pseudo-op, 6-21
- /VOLIDS switch,
 - DUMPER, 7-5
- Volumes in a set,
 - Switching between, 7-21
- /W switch,
 - CREF, 5-4
 - FILCOM, 4-8, 4-11
- Warning messages,
 - CREF, 5-10
 - DUMPER, 7-57
 - MAIL, 2-6

INDEX (CONT.)

Warning messages (Cont.)
 MAKLIB, 6-28
 PLEASE, 8-6
 RDMAIL, 3-7
WHEEL capabilities, 2-5,
 3-2, 7-1
/WHO switch,
 MAKLIB, 6-6, 6-25, 6-26,
 6-27
/WHO switch example,
 MAKLIB, 6-26
WILD program, 6-28
Wildcards, 1-2
 DUMPER, 7-23
/WRITE-ENABLED switch,
 DUMPER, 7-4

Writing and reading a
 TOPS-10 tape, 7-16
Writing tape in 36 bits,
 Reading and, 7-16

/X switch,
 FILCOM, 4-8, 4-11

/Z switch,
 CREF, 5-4

READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

Did you find errors in this manual? If so, specify the error and the page number.

Please indicate the type of reader that you most nearly represent.

- ☐ Assembly language programmer
- ☐ Higher-level language programmer
- ☐ Occasional programmer (experienced)
- ☐ User with little programming experience
- ☐ Student programmer
- ☐ Other (please specify) _____

Name _____ Date _____

Organization _____ Telephone _____

Street _____

City _____ State _____ Zip Code _____
or
Country

Please cut along this line.

----- Do Not Tear -- Fold Here and Tape -----

digital



No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL

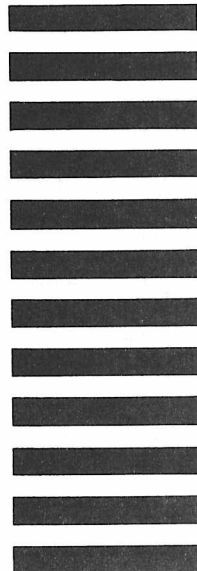
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

SOFTWARE PUBLICATIONS

200 FOREST STREET MR1-2/E37

MARLBOROUGH, MASSACHUSETTS 01752



----- Do Not Tear -- Fold Here and Tape -----

Cut Along Dotted Line

READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

Did you find errors in this manual? If so, specify the error and the page number.

Please indicate the type of reader that you most nearly represent.

- ☐ Assembly language programmer
- ☐ Higher-level language programmer
- ☐ Occasional programmer (experienced)
- ☐ User with little programming experience
- ☐ Student programmer
- ☐ Other (please specify) _____

Name _____ Date _____

Organization _____ Telephone _____

Street _____

City _____ State _____ Zip Code _____
or Country

-----Do Not Tear - Fold Here and Tape-----

digital



No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL

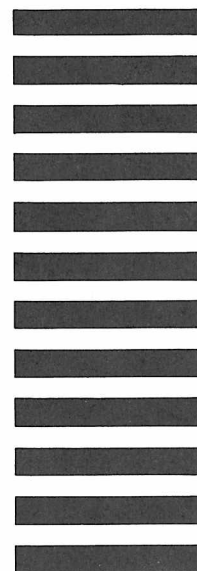
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

SOFTWARE PUBLICATIONS

200 FOREST STREET MR1-2/L12

MARLBOROUGH, MASSACHUSETTS 01752



-----Do Not Tear - Fold Here and Tape-----

Cut Along Dotted Line

READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

Did you find errors in this manual? If so, specify the error and the page number.

Please indicate the type of reader that you most nearly represent.

- ☐ Assembly language programmer
- ☐ Higher-level language programmer
- ☐ Occasional programmer (experienced)
- ☐ User with little programming experience
- ☐ Student programmer
- ☐ Other (please specify) _____

Name _____ Date _____

Organization _____ Telephone _____

Street _____

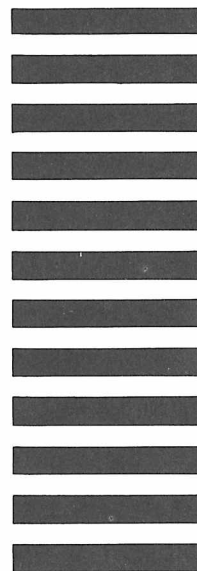
City _____ State _____ Zip Code _____
or Country

-----Do Not Tear - Fold Here and Tape-----

digital



No Postage
Necessary
if Mailed in the
United States



BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

SOFTWARE PUBLICATIONS
200 FOREST STREET MR1-2/L12
MARLBOROUGH, MASSACHUSETTS 01752

-----Do Not Tear - Fold Here and Tape-----

Cut Along Dotted Line

YX 2320

Reg. Res.
LCB

UPDATE NOTICE

TOPS-20 User Utilities Guide AD-D850A-T1

January 1982

Insert this Update Notice in the *TOPS-20 User Utilities Guide* to maintain an up-to-date record of changes to the manual.

Changed Information

The changed pages contained in this update package reflect an updated version (2B) of the MAKLIB utility.

The instructions for inserting this update start on the next page.

Copyright ©, 1982, Digital Equipment Corporation.
All Rights Reserved.

INSTRUCTIONS

AD-D850A-T1

The following list of page numbers specifies which pages are to be placed in the *TOPS-20 User Utilities Guide* as replacements for, or additions to, current pages.

[Title page
Copyright page

[6-3
6-4

[6-25
6-30.1

[6-33
6-40

[6-43
6-45

KEEP THIS UPDATE NOTICE IN YOUR MANUAL TO MAINTAIN AN UP-TO-DATE RECORD OF CHANGES.

TYPE AND IDENTIFICATION OF DOCUMENTATION CHANGES.

Five types of changes are used to update documents contained in the TOPS-20 software manuals. Change symbols and notations are used to specify where, when, and why alterations were made to each update page. The five types of update changes and the manner in which each is identified are described in the following table.

The Following Symbols and/or Notations

1. Change bar in outside margin; version number and change date printed at bottom of page.
2. Change bar in outside margin; change date printed at bottom of page.
3. Change date printed at bottom of page.
4. Bullet (●) in outside margin; version number and change date printed at bottom of page.
5. Bullet (●) in outside margin; change date printed at bottom of page.

Identify the Following Types of Update Changes

1. Changes were required by a new version of the software being described.
2. Changes were required to either clarify or correct the existing material.
3. Changes were made for editorial purposes but use of the software is not affected.
4. Data was deleted to comply with a new version of the software being described.
5. Data was deleted to either clarify or correct the existing material.

January 1982



tops20

digital

digital

digital equipment corporation